INDEKS Nr 32034 X ISSN 1230-7947

Commodore

1/1994

AMIGOWIEC

Pismo użytkowników komputerów AMIGA

cena 25.000 zł



Karta grafiki 24-bitowej
EGS 28/24
SPECTRUM

Oprogramowanie:

- ✓ TYPE SMITH
- ✓ REAL 3D v2.0
- ✓ FINAL WRITER
- ✓ OPUS 4

WARSZTATY!

- AMOS
- Arexx cz. 2
- Repro Studio
- Imagine cz. 1
- Term

Ciąg dalszy na stronie 75

AMIGOWIEC 12/92



PD #12/92 Pakiet Gier

Dysk A

AMIGOWIEC 1-2/93

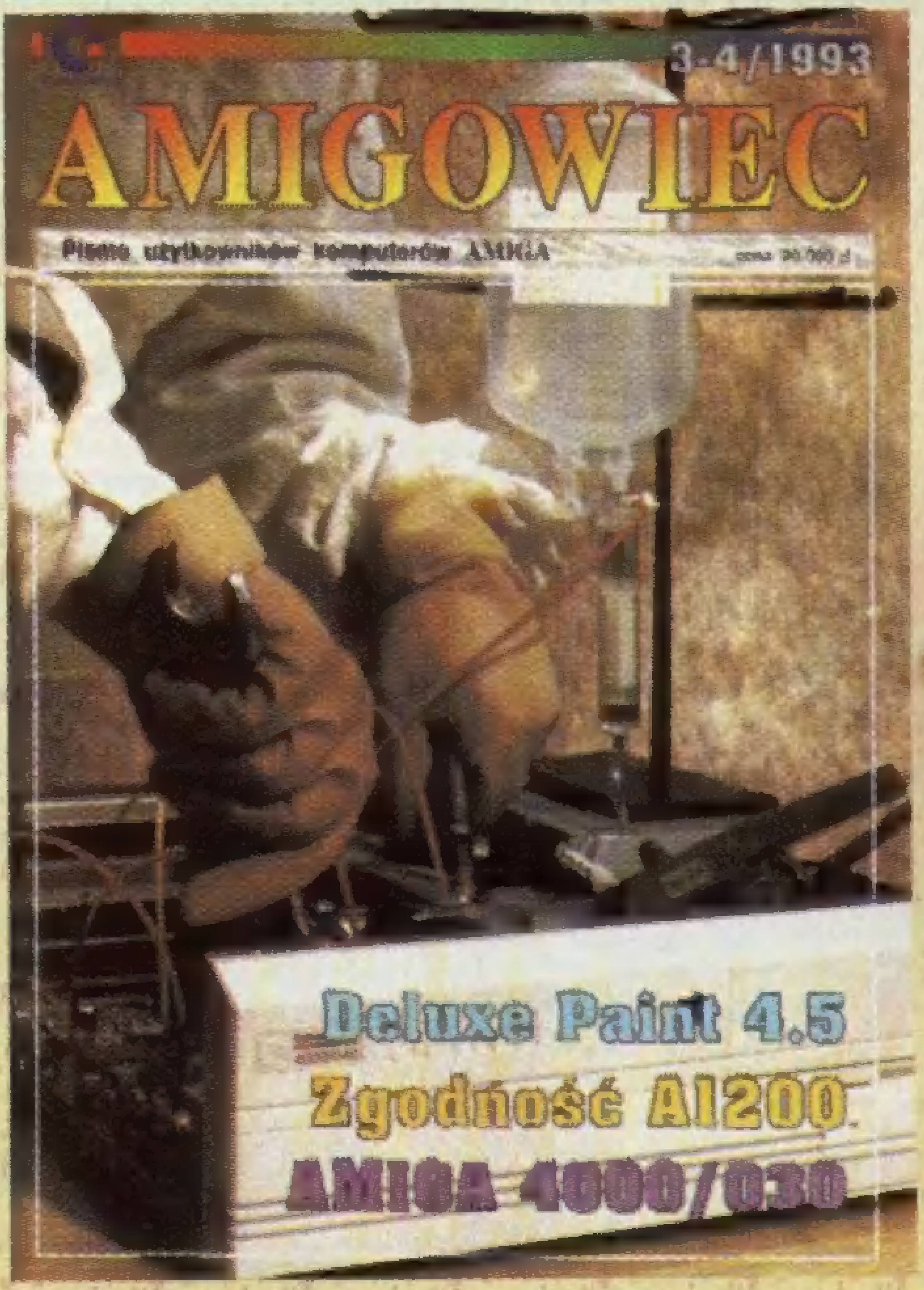


PD #1/93

PD #2/93

PD C

AMIGOWIEC 3-4/93



PD #3-4/93

Dysk B

AMIGOWIEC
ROK 5 NR 1/94 (35) INDEX 32034X

REDAKCJA

REDAKTOR NACZELNY:
RYSZARD KOWALSKI

Z-CA REDAKTORA NACZELNEGO:
KRZYSZTOF NOWICKI

SEKRETARZ REDAKCJI:
TOMASZ ŁOBODA

OPRACOWANIE GRAFICZNE:
KRZYSZTOF WIRSZYŁŁO
BARTŁOMIEJ WRÓBLEWSKI

SKŁAD KOMPUTEROWY:
MAŁGORZATA LEWANDOWSKA

KOREKTA:
KAROL JAGODZIŃSKI
ŁUCJA UMIŃSKA

KOLPORTAŻ:
ANDRZEJ KENTZER

PRENUMERATA:
AGNIESZKA PRZYBYLSKA

WSPÓŁPRACA:
WOJCIECH BIAŁKOWSKI,
WOJCIECH CZYŻ,
JAROSŁAW CHROSTOWSKI,
MARCIN GACKOWSKI,
ADAM GREGROWICZ,
TOMASZ FLANC,
TOMASZ HRYCUNIAK,
MACIEJ KLIMKIEWICZ,
TOMASZ KOKOSZCZYŃSKI,
SEBASTIAN KŁOMSKI,
TOMASZ KULBACKI,
MARIUSZ ŁAWICKI
TOMASZ ŁOBODA,
ARTUR ŁUKASIK,
TOMASZ MATAJEK,
IZABELA SKIBIŃSKA,
MAREK STOR,
KRZYSZTOF TROJANOWSKI,
DARIUSZ TRZOS

ADRES REDAKCJI:
ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. (+52) 28-79-20, fax (+52) 22-64-03

KONTO BANK PKO S.A. BYDGOSZCZ
00509011-04009339-2511-30-00111o

SKŁAD NA KOMPUTERACH AMIGA
OKŁADKA: WOJCIECH BIAŁKOWSKI
DRUK: "ZAKŁADY GRAFICZNE" SP. Z O.O.
UL. OKRZEI 5, 64-920 PIŁA

Sensacje i opowieści



Witam wszystkich, którzy zechcieli sięgnąć po kolejny numer naszego pisma. Na początek kilka uwag odnośnie przeszłości. Niestety, w numerze świątecznym nie uniknęliśmy paru wpadek. Tym razem chochlik drukarski nie tylko okazał się złośliwy, ale i bezlitosny. Wnikliwie śledząc nasze poczynania, uderzał tam, gdzie najmniej się tego spodziewaliśmy (zgodnie z prawami Murphy'ego). Umiejscowienie drania kosztowało nas (mnie osobiście także) wiele sił witalnych i mamy nadzieję, że lekcję jaką otrzymał, zapamięta długo.

Nie spełniły się jednak moje zapowiedzi odnośnie terminu ukazania się Amigowca 1/94. Ukazujemy się w lutym i jest to frycowe za zmiany organizacyjne pisma. Rekompensujemy to zwiększeniem ilości stron kolorowych, a to jeszcze nie koniec zmian. Numer 2/94 ukaże się około 15 marca. W wakacje planujemy numer podwójny. Po wakacjach numeracja zgodna z miesiącami.

Aby rozwiać wszelkie wątpliwości informuję, że zgodnie z wcześniejszymi zapowiedziami cena prenumeraty Amigowca wynosi 138.000 zł za 6 wydań i odpowiednio 256.000 zł - za 12 wydań. Cena dyskietek PD Amigowca wynosi w wolnej sprzedaży 48.000 zł, a dla prenumeratorów 38.000 zł.

Kontynuując wątek pesymistyczny naszego spotkania, pragnę poinformować o likwidacji założonego w 1992 roku biura przedstawicielskiego Commodore w Polsce. Nie oznacza to upadku firmy, jak niektórzy mogliby wnioskować, a jedynie przeniesienie ośrodka decyzyjnego do Frankfurtu. Komentarz w redakcji był jeden "(...) a właściwie to było co likwidować?". Szerszą analizę tego faktu przeprowadzimy w jednym z kolejnych numerów Amigowca.

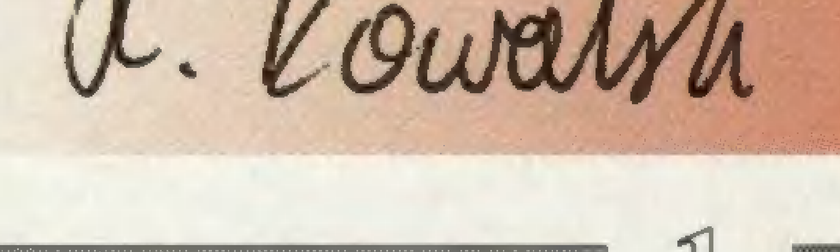
No, ale dość pesymizmu. Parę dobrych wiadomości powinno Was trochę rozchmurzyć. W momencie, gdy czytacie te słowa zostało już uchwalone przez Sejm RP nowe prawo autorskie (czekamy na wejście w życie). I co, jak się czujecie? Idąc za ciosem, pragnę donieść, że nasz wydawca podpisał umowę licencyjną ze znaną amerykańską firmą Soft-Logic Publishing, najmocniejszym producentem oprogramowania desktop-publishing na Amigę. Zaowocuje to wydaniem w przeciągu nadchodzącego roku następujących pozycji: Page Stream 3.0 - program do komputerowego składu czasopism, Type Smith 2.0 - edytor fontów wektorowych, Art Expression - wektorowy program graficzny. Oczywiście, wszystkie produkty będą w pełni po polsku. Obecnie na tapecie znajduje się Type Smith 2.0, którego wydanie planowane jest na marzec. Równocześnie powinna pojawić się (w końcu) polska wersja Workbencha, lecz w tym przypadku zależy to szczególnie od prężności Commodora. O postępach w pracach będziemy Was informować na bieżąco.

Co w numerze? Na uwagę zasługuje TRYBUNA, która dzięki mojemu apelowi wróciła na łamy pisma. Dzisiaj krótka rozprawa o Cyberpunku. Zresztą co niektórzy członkowie redakcji bardzo napalili się na tę rubrykę. A Wy nie? Cały czas czekamy na Wasze listy, wnioski, uwagi, opinie, fakty, to wszystko, co stanowi treść TRYBUNY. Uff, ale oficjalnie.

Oprócz tego w numerze: kurs Imagine - nasze nowe dziecko, zobaczymy jak się sprawdzi; kontynuacje rozpoczętych kursów Assemblera, Amosa, Arexxa; artykuły m. in. o EGSie, Opusie, Realu, Final Writerze. Zresztą, zobaczcie sami.

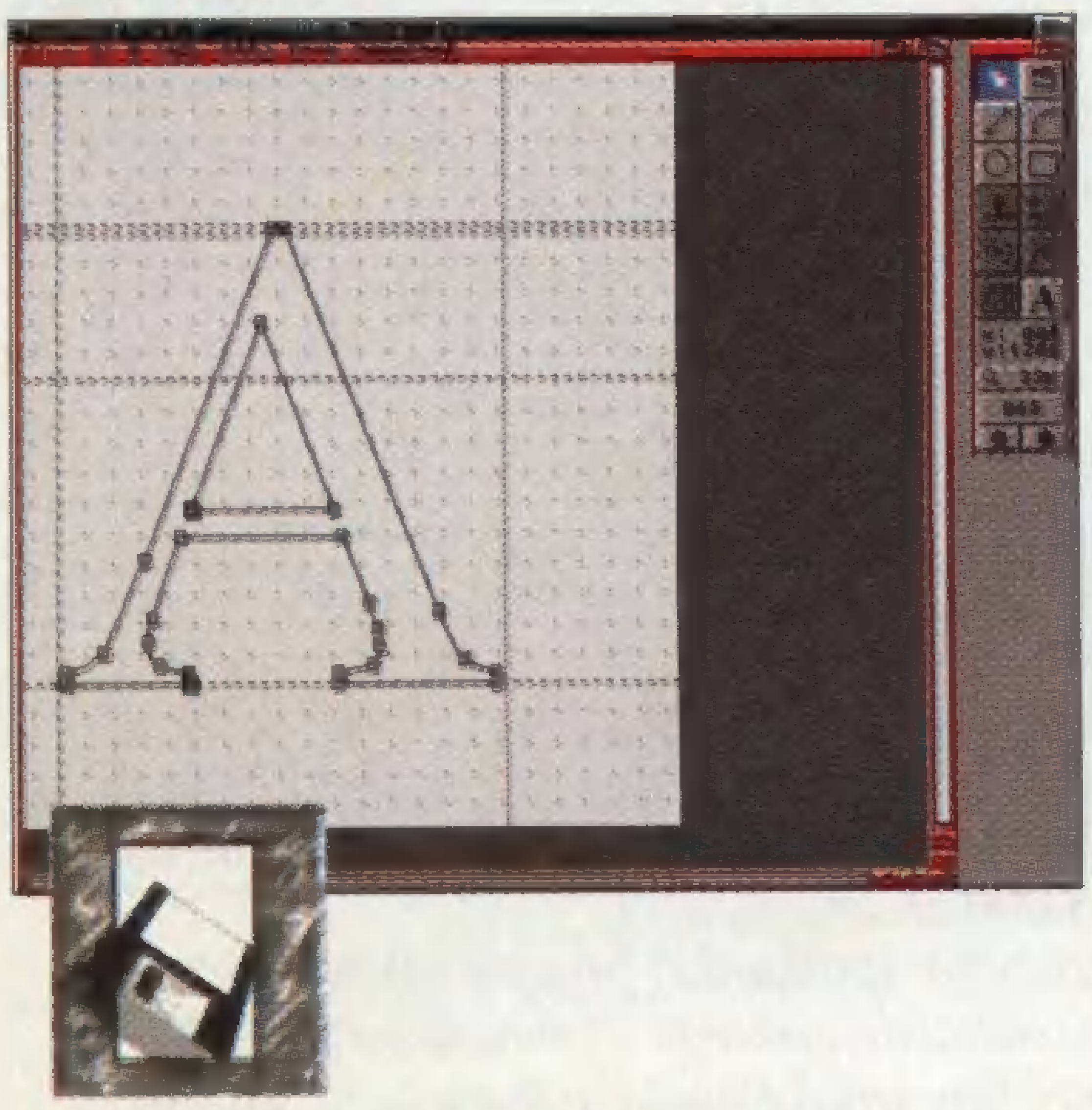
Na koniec uchylę rąbka tajemnicy o naszej wydawniczej przyszłości - planujemy przeprowadzenie ankiety (z ciekawymi nagrodami). Już dzisiaj ostrzcie ołówki i analizujcie, co Wam się podoba, a co nie. Bez Waszej pomocy to pismo będzie tylko martwym kawałkiem papieru.

Do zobaczenia za miesiąc.





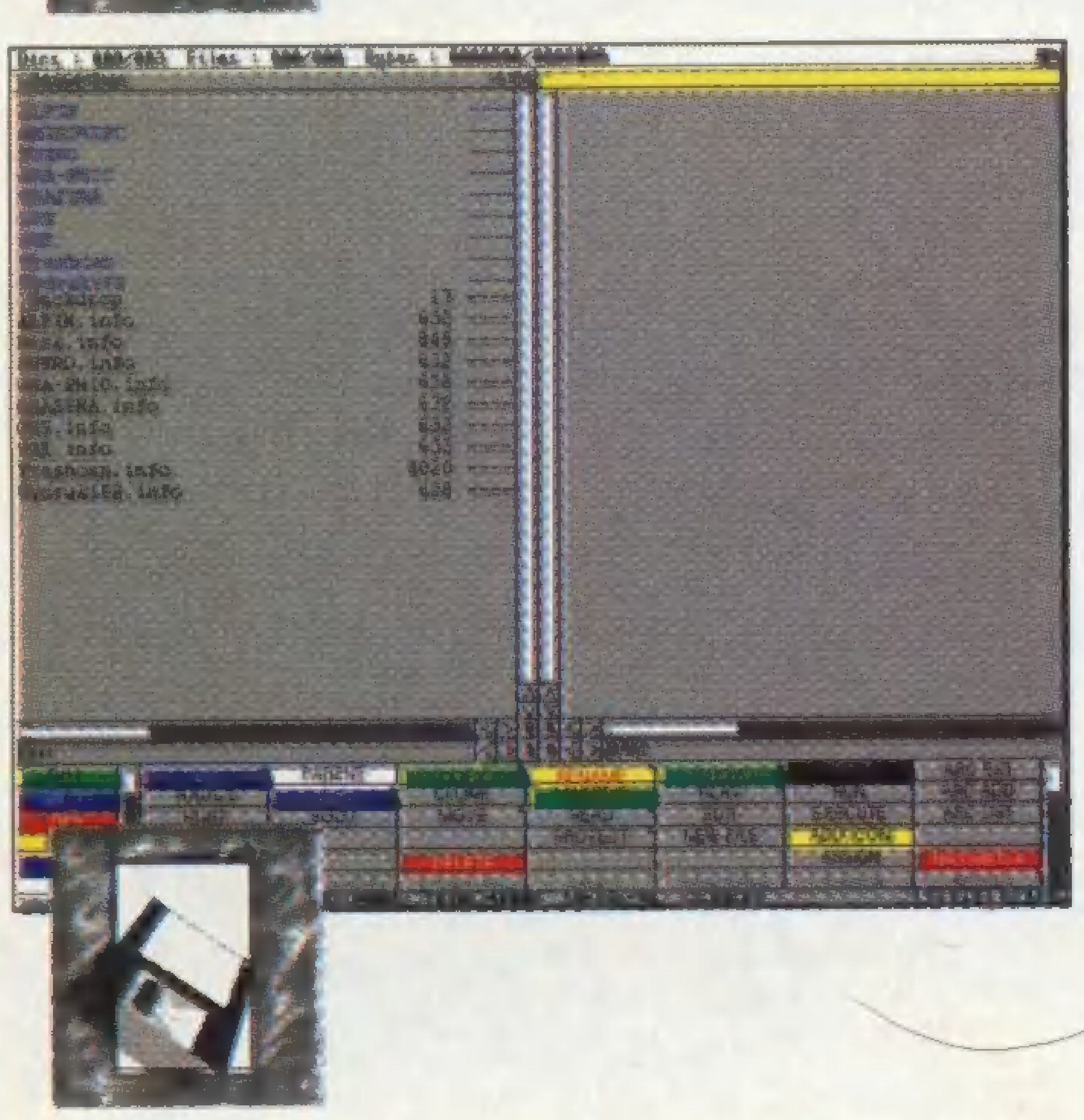
10 **EGS**
Tomasz Matajek prezentuje najnowsze graficzne dziecko firmy GVP. Nowa jakość w temacie grafiki.



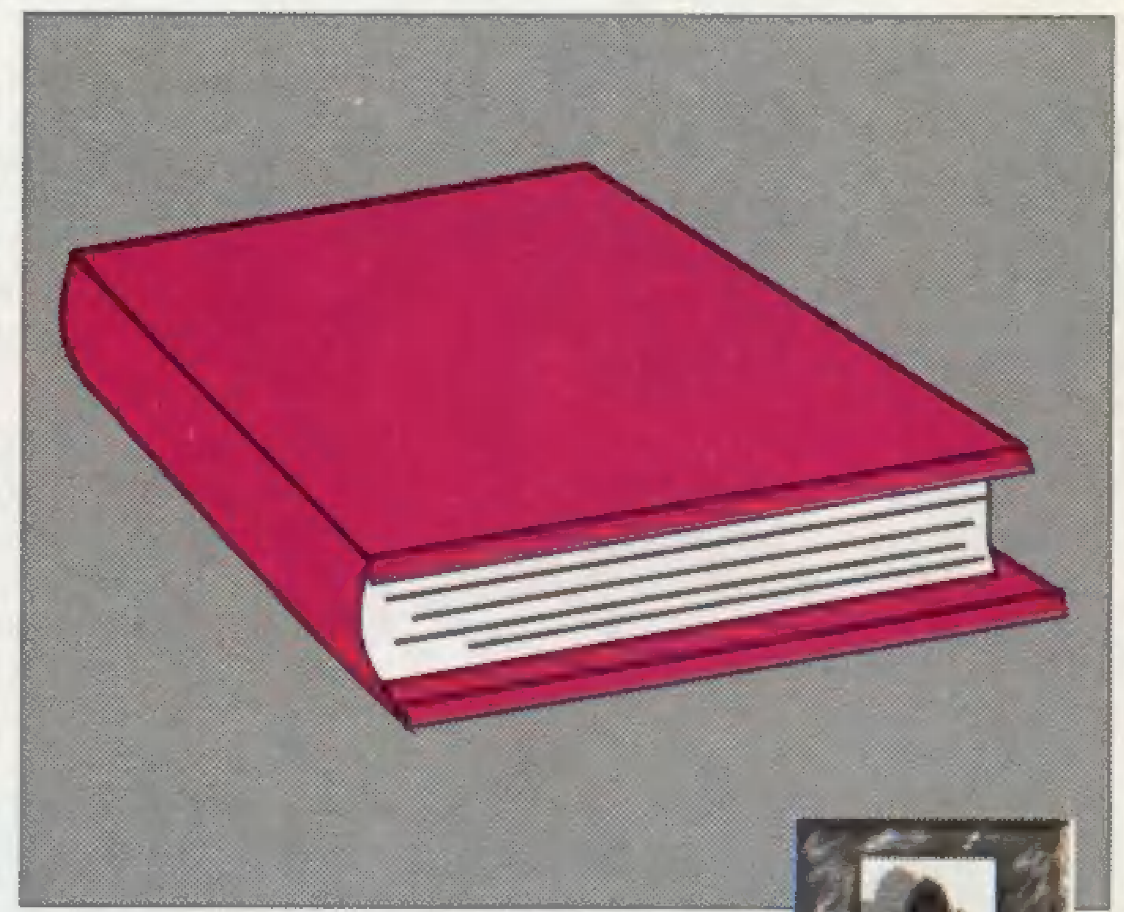
13 **TYPE SMITH**
Przedstawienie edytora czcionek wektorowych. W jednym z najbliższych numerów uzupełnienie o wersji 2.0.



15 **REAL 3D V2.0**
Program, który jest konkurencją dla stacji graficznych. Krzysztof Trojanowski przedstawia możliwości programu.

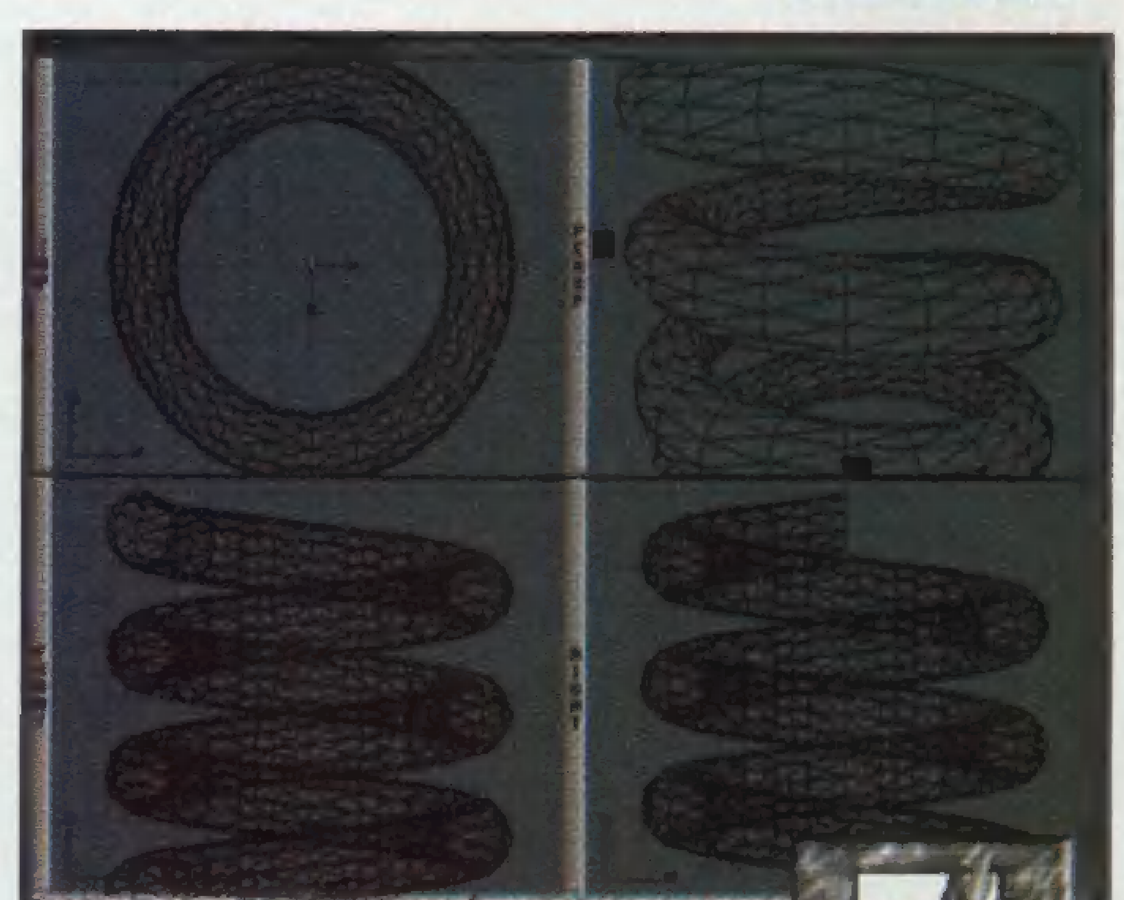


22 **OPUS V4.11**
Nowe wcielenie giganta opracowane przez Tomka Kokoszczyńskiego. Zobaczcie sami.





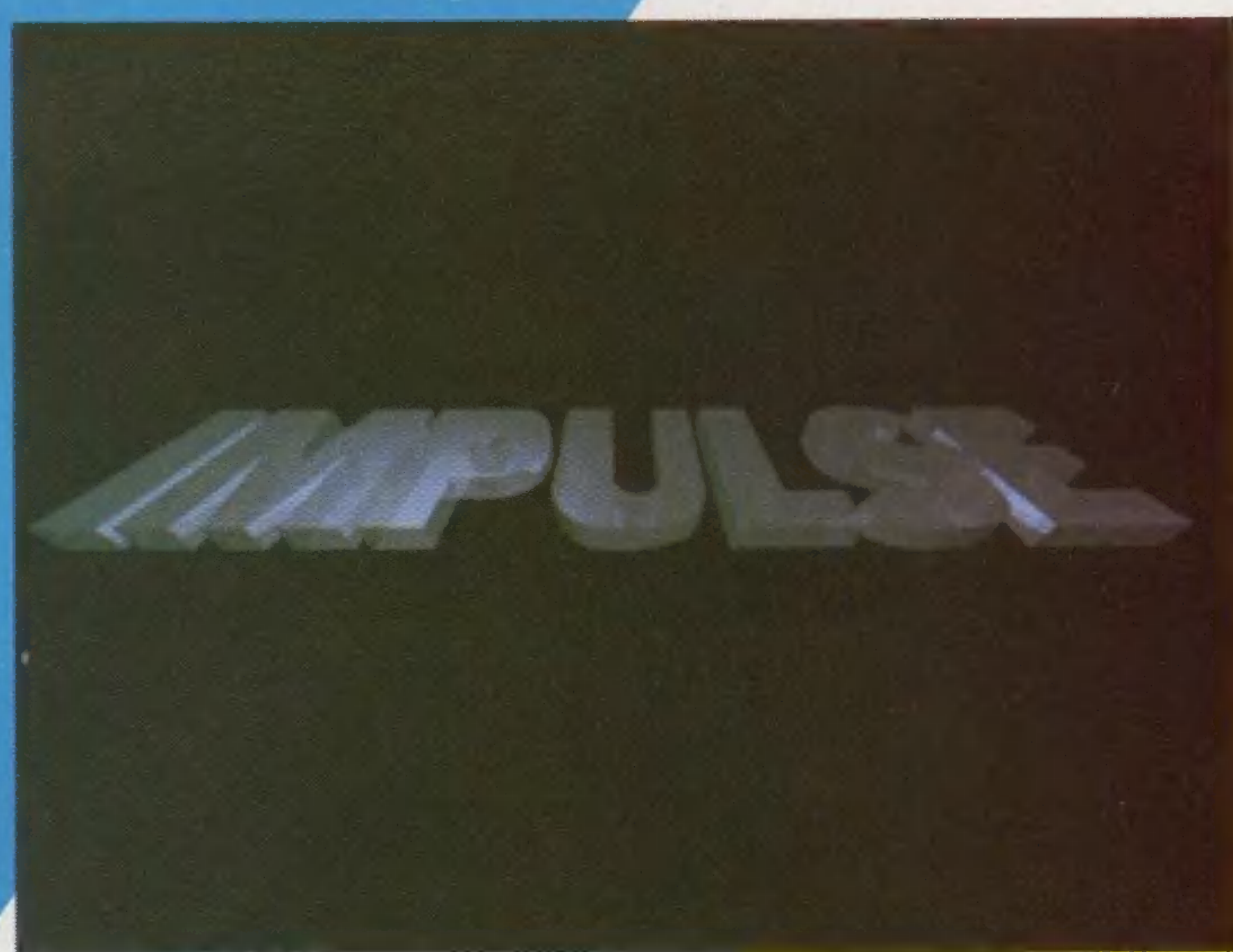

CYBER




KISS ME!






IMPULSE






FACET






Suny


WALT.





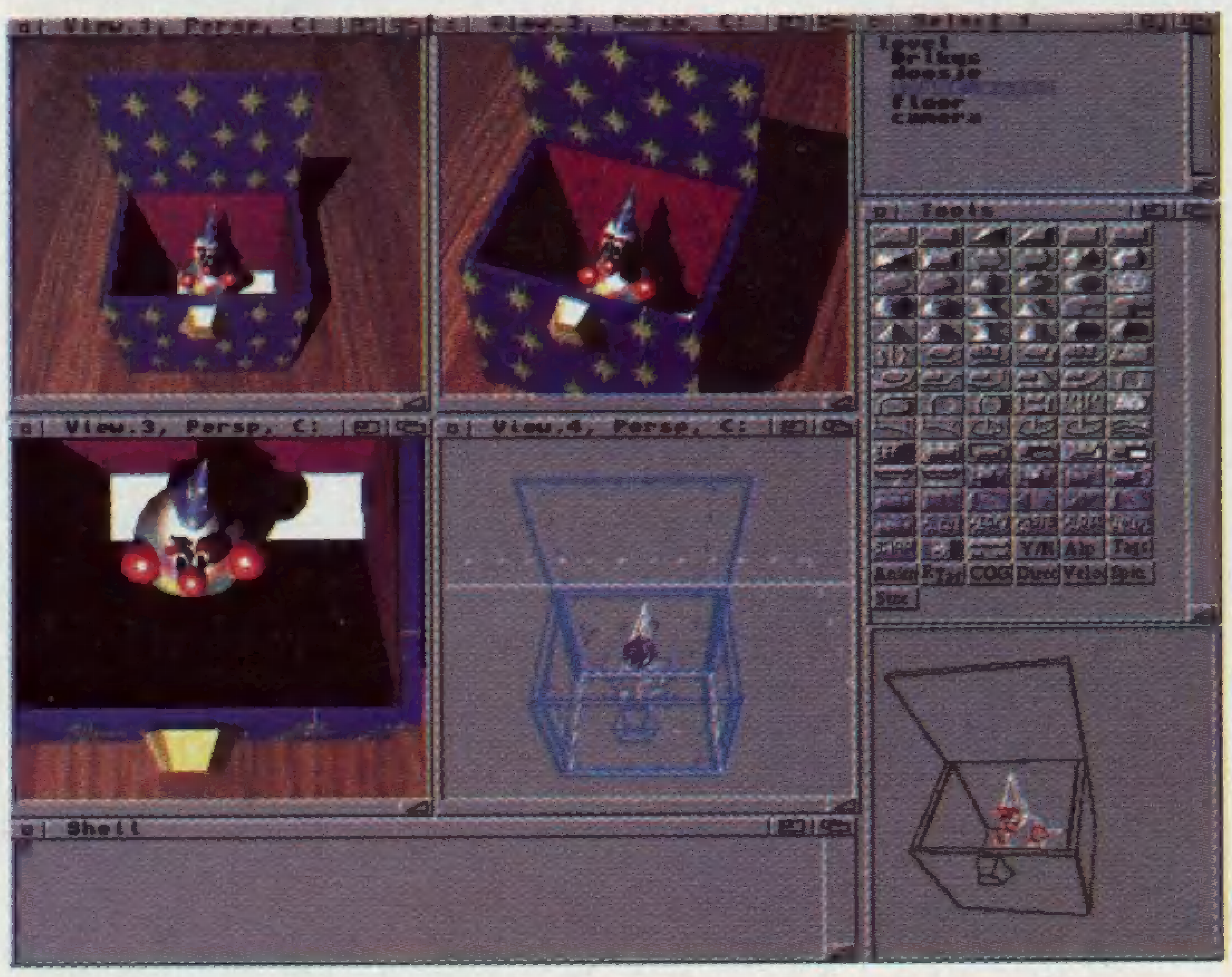


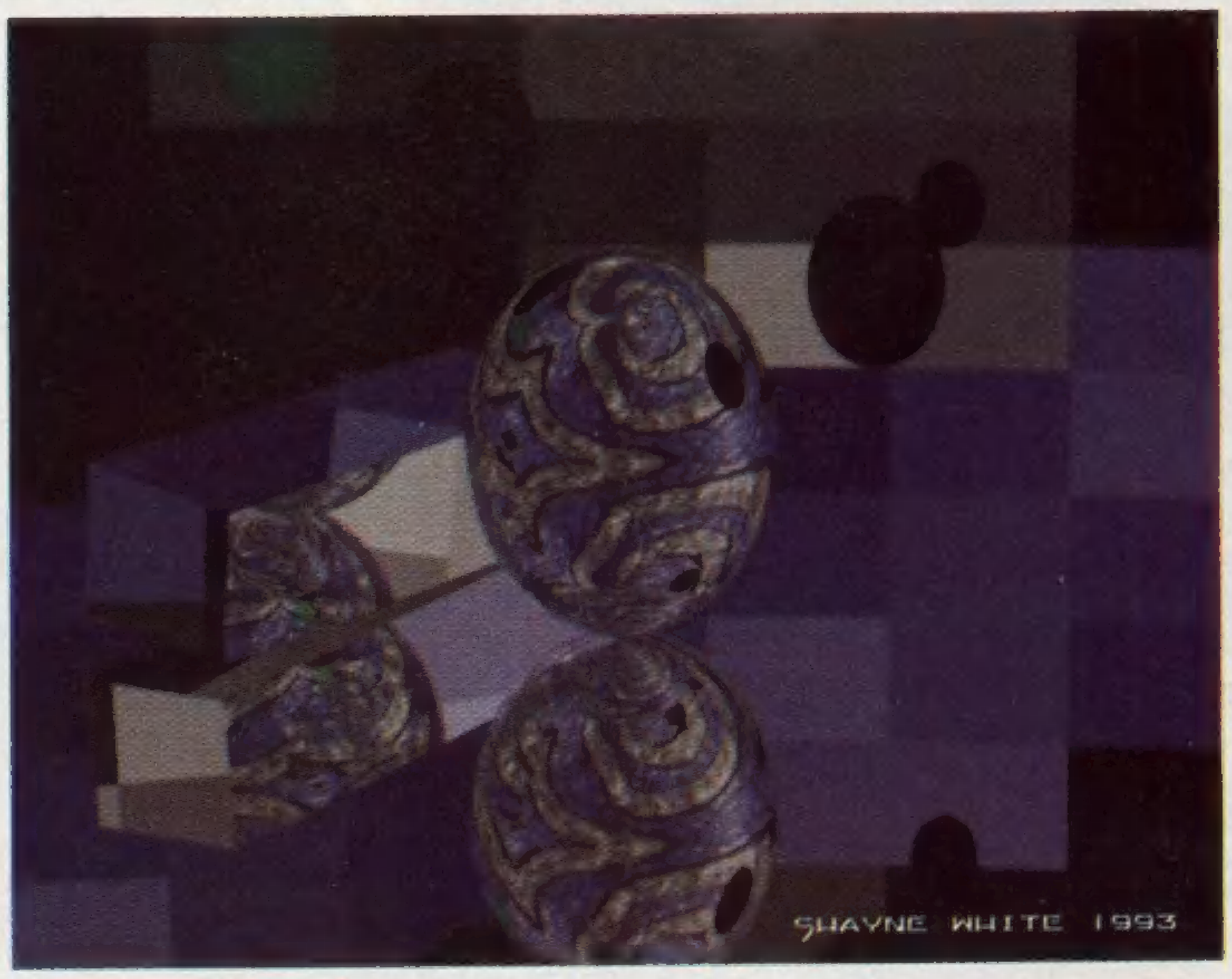











SWAYNE WHITE 1993

DEPESZE

Tomasz Kokoszczyński, Ryszard Kowalski

PCMCIE do boju!

Specyficzny port Amigi 1200 określany jako PCMCIA (potocznie: pcimcia) do tej pory spełniał jedynie zadanie otworu wlotowego dla kurzu. Powoli jednak zaczyna się pojawiać coraz więcej przydatnych dodatków, które można podłączyć właśnie do tego portu. O ile podłączenie rozszerzenia pamięci jest na tyle drogie, że aż nieopłacalne, o tyle może inne produkty będzie łatwiej (czytaj: taniej) tam umieścić. Pojawiły się już modemy, karty sieciowe, a nawet dyski twarde (patrz. Krótkie opisy), które można tam wetknąć. W najbliższym czasie (do Świąt Bożego Narodzenia) ma się pojawić 16-bitowy sampler podłączalny do tego złącza. Produkt jest sygnowany przez Great Valley Products - znaną amerykańską firmę. Do osprzętu ma być dołączany dysk z bogatym oprogramowaniem. Pożyjemy, zobaczymy.

Domowy edytor video

Entuzjaści video posiadający Amigę dostaną wkrótce do ręki (za niewielką opłatą 199 funtów) program do edycji domowych filmów i dokładania do nich ścieżek dźwiękowych: Edit Mate. W zestawie znajdzie się odpowiednie oprogramowanie, kabelki i stereofoniczny mikser dźwięku. Zestaw pozwoli na edycję i zapisywanie w dowolnym porządku do 99 scen. Dla tych, którzy będą mieli jakieś problemy z obsługą przewidziano pomoc typu "On line".

Amigowe PCty taniej?

Każdy z amigowców z pewnością zetknął się z problemem przenoszenia danych z IBMa, a jeszcze lepiej z próbą uruchomienia IBMowskich programów na Amidze. Powstało wiele emulacji tego komputera, a najbardziej efektywna z nich nosi nazwę PC Task. Program ten oczywiście umożliwia bezproblemowe przenoszenie plików z Amigi na IBMa i na odwrót. Możliwe jest także podłączenie dysku o wysokiej gęstości zapisu używanego jako dysk IBMa, jak i wykorzystanie dysku twardego. Program emuluje następujące karty graficzne "Błękitnego": MDA, CGA, EGA i VGA. W tej ostatniej możliwe jest wyświetlanie do 256 kolorów. Szybkość działania jest zależna od szybkości Amigi, na której uruchomimy program.

Chciałbym zwrócić jednak Państwa uwagę na drobny fakt, że o ile spora część programów z IBOLa może całkiem nieźle działać, to jak na razie należy zarzucić marzenia o sprawnym (tzn. o jakimkolwiek) funkcjonowaniu systemu Windows na Amidze, a co za tym idzie wszystkich aplikacji działających pod tym systemem. Za cenę 50 funtów dostajemy tylko pewną część IBMa i jak na razie, twórcom oprogramowania jest jeszcze daleko do pełnej emulacji, którą skutecznie można wykonywać jedynie przy użyciu odpowiedniego osprzętu. No cóż, być może w bliższej bądź dalszej przyszłości...

Co nowego w Commodorze?



Jak wszyscy wiecie Commodore jest firmą, która od zawsze zajmuje się produkcją naszej kochanej przyjaciółki. W Europie przyzwyczailiśmy się do tego, że "jakaś tam" siedziba firmy znajduje się we Frankfurcie w RFN. To nie tak hop. Owszem w RFN jest filia firmy - jednak główny jej oddział mieści się w Stanach Zjednoczonych Ameryki. Ostatnio okazało się nawet, że pod latarnią najciemniej i w USA rynek Amigi nie tylko nie rozkwita, ale wręcz zanika. Szef firmy Jim Dionne żartuje sobie nawet, że Commodore toruje drogę innym firmom ogłaszając straty i restrukturyzację przedsiębiorstwa. Żarty jakieś takie smutne. Jest rzeczą oczywistą, że wielkie firmy w pewnym momencie przestają sobie z samymi sobą dawać radę. Jednak jest to nikłe pocieszenie dla użytkowników Amig. Oczywiście szefowie firmy nie są gapami karmieni i podejmują różne środki zaradcze. Jednym z nich jest oferta licencyjna produktów firmy. Np. oddaje się w "dzierżawę" opracowanie i sprzedaż karty usuwającej interlace w Amigach wyposażonych w kości AGA. Do dyspozycji firm, które mają na to ochotę stoi również karta łącząca Amigi z Ethernetem.

Oczywiście są też dobre wiadomości z firmy: nowy układ graficzny AAA wchodzi właśnie w fazę testów i mimo, że jest objęty "szczelną" tajemnicą to można się po nim spodziewać wielu rewelacji (16 mln kolorów bez picu) i chyba niestety wielu niekompatybilności. Nie wiadomo, czy możliwość wstawienia nowych układów do A4000 pozostanie tylko w sferze plotek, czy też będzie z tego coś więcej. Poza tym firma chce ponownie opanować rynek amerykański przy pomocy swojej maskotki opisywanej w ostatnim numerze AMIGOWCA: CD32.

Siłą wspierającą dla AMIGI mają być: wspomniany w ostatnich depeszach program PageStream 3.0 i opisany w ostatnim AMIGOWCU Emplant 3.15, emulujący MacIntosha w systemie 7.x, a także podobno w najbliższej przyszłości IBMa 486. Poza tym firma GVP "ulitowała" się nad wszystkimi posiadaczami Amig 1200 i stworzyła kartę dopalacza bazującą na pełnej wersji procesora 68030 (patrz poprzedni AMIGOWIEC). Gdy słowo stanie się faktem, to rzeczywiście konkurencja Amigi będzie wyglądała blado. Na razie jednak Commodore przeżywa kłopoty, a to jaka może być jedna z ich przyczyn



prawdopodobnie opiszemy w najbliższym czasie.

AMOS podbija świat

AMOS-Basic jest jednym z bardziej popularnych języków programowania na Amigę. Język ten jest prosty w użyciu (Basic) i ma znakomicie opracowany system obsługi programowania (AMOS). W obecnym numerze AMIGOWCA możecie skorzystać z 5 odcinka AMOSOWEGO kursu. Na Zachodzie oprócz dodatków typu: kompilator, pojawia się bogata literatura dotycząca tego języka. Ostatnio nawet jest dostępna książka "AMOS in Education", która podpowiada jak stosować AMOSa przy tworzeniu programów edukacyjnch. Oczywiście książka jest w języku angielskim. Redakcji AMIGOWCA wydaje się, że upowszechnienie tak wspaniałego komputera jakim jest Amiga, a wraz z nią tego prostego i wygodnego języka przyniosłoby większe efekty w naszym systemie edukacyjnym, niż niejedna "huczna" reforma szkolnictwa, z której potem nic nie wynika.

Rodzice do pracy, dzieci do AMOSA!

Błyszczące pieniądze

Jak robić pieniądze z niczego możemy się dowiedzieć z różnych gazet, w których np. jeden pan przekonuje nas, że on to wie i wystarczy mu przysłać coś koło 159 tys., by się dowiedzieć, że my też to możemy robić tak samo i do szczęścia nic nam nie będzie potrzebne. To się teraz nazywa "tajemnica". Inną politykę robienia pieniędzy praktycznie z niczego prowadzi firma Berkeley Systems, która zajmuje się między innymi tworzeniem "oszczędzaczy" ekranów (tzw. saverów). Są to śmieszne obrazki i inne mrugadełka, które pojawiają się w czasie, gdy my oderwiemy się na chwilę od naszego ulubionego komputera. Wprawdzie oryginalne systemy Amigi od 2.x wzwyż oferują Blankera, który zajmuje się rysowaniem różnogiętych kresek na czarnym tle, ale firma Berkeley wypuściła na rynek coś o wiele ładniejszego. Jest ona znana z oszczędzania monitorów systemu MacIntosh i ma w tej dziedzinie spore tradycje. Oczywiście za te dodatki trzeba dodatkowo płacić. Ludzie czynią to chętnie, bo każdy lubi, gdy jego komputer sam bawi się ze sobą. W ten prosty sposób firma zarobiła w ubiegłym roku na tego typu Nikomu_Nie _Potrzebnych rzeczach około 18 mln $. To się nazywa Ameryka!

Przyspieszenie z Hypercache'a

HyperCache Professional jest przyspieszaczem różnych amigowych urządzeń. Ta prosta zabawka może przyspieszyć Wasz twardy dysk do 22 razy, a zwykłą stację do 320 razy. HyperCache działa ze wszystkimi urządzeniami, które zapisują dane w systemie blokowym. Przy wykorzystaniu specjalnego algorytmu, o dźwięcznej nazwie LRU, "dopalarka" zapamiętuje pewną ilość danych odczytanych z innego urządzenia i w momencie, gdy system się ich domaga, pozwala je szybciej przekazywać. To cudeńko kosztuje 50 funtów i działa na wszystkich Amigach. Niestety redakcja nie posiada bardziej szczegółowych danych na temat tej rewelacji. Producentem urządzenia jest firma Silicon Prairie Software, a dystrybutorem firma Meridian Software Distribution.

Nowa AMIGA 4000T!



Firma Commodore zaprezentowała na targach World of Commodore Show, które odbyły się w Pasadenie najnowszą z rodziny Amig: 4000T, gdzie T oznacza oczywiście Tower (wieża). Producenci twierdzą, że to nie jest wieża, bo nazwa Tower jest zastrzeżona i nie może być używana w nazwach komputerów. No, ale literka "T" może być... Amiga ta będzie w swej konstrukcji oczywiście oparta na Amidze 4000 z procesorem 68040 25MHz. Podstawową zaletą nowej Amigi jest oczywiście jej pojemność - zmieści się w niej o wiele więcej kart i innych rozszerzeń niż w "standardowej" A4000. Z tego też powodu dodano kilka złączy. Jenym z nich jest dodatkowe złącze video, tak że teraz możemy naszą przyjaciółke wyposażyć w dwie karty graficzne np. w Toastera i OpalVision. Nie żałowano również na złączach Zorro III i nowa Amiga ma ich aż 5. Wszyscy z utęsknieniem patrzący na dyski twarde typu SCSI doczekali się ich kontrolera i to aż SCSI 2, który ma o wiele większą szybkość przenoszenia danych niż zwykły SCSI. Kilka dysków z przodu komputera, zamienionych ewentualnie na CD-ROM, czy TapeStreamer może dopełnić całości. W tej chwili projekt nowej A4000T jest już zamknięty i powinna się ona pojawić na rynku na początku 1994 roku. Jedyną przeszkodą może być Amiga CD32, która jest produkowana w tej samej fabryce leżącej na Filipinach. Jak na razie zabawki mają w naszym wspaniałym świecie wyższy priorytet od poważnego sprzętu. Zdziwienie może budzić tylko fakt, dlaczego nowej Amigi nie wyposażono w jeszcze szybszy procesor z rodziny 68040? Takie procesory taktowane zegarem 40MHz są przecież łatwo dostępne na rynku. Commodore na ten temat milczy...

Amiga C Club

Amiga C Club został założony przez młodego Szweda, studenta ekonomii, aby służyć pomocą wszystkim amigowcom zajmującym się programowaniem w języku C. Istnieje od 3 lat i ma ponad 200 członków z 38 krajów. Działalność tego klubu polega na wydawaniu i zbieraniu informacji do nowych wydań AMIGA C Encyklopedia oraz udzielaniu pomocy dla zarejestrowanych członków tego klubu. Aktualna wersja encyklopedii lub raczej podręcznika języka C nosi numer 3.0 i mieści się na 12 dyskach (10 public domain i 2 dla zarejestrowanych członków). Dodatkowo w przygotowaniu jest nowa wersja podręcznika (4.0), napisana od nowa specjalnie pod kątem systemu 2.x. Przykłady tego podręcznika są kompatybilne z kompilatorami SAS 6.xx, Manx Aztec 5.xx, Lattice4.xx, itp. Zarejestrowani członkowie mają nieograniczony dostęp do dokumentacji, encyklopedii i BBSu. Na swoje pytania i problemy związane z programowaniem w C otrzymują odpowiedzi bezpłatnie. Poza tym mogą używać w swoich programach (komercyjnych i niekomercyjnych) bogaty zasób kodów źródłowych z biblioteki klubu. Mogą oni bezpłatnie zeskanować zdjęcia (skanerem Sharp JX-100 Colour) o rozmiarach nie przekraczających 10x15 cm oraz tworzyć sample (Future Sound Sampler przy sample rate do 42,118 sampli /sekundę. Wszystkie przesyłki przesyłane są pocztą lotniczą pierwszej klasy (polecone). Istnieje od pewnego czasu BBS otwarty dla wszystkich, jednak dla użytkowników spoza klubu o znacznie ograniczonym dostępie. Działa on na A2000C z modemem SupraFax 14.400 bps V32bis (numer: +48 8 765 45 85). Opłata dająca wieczyste członkowstwo wynosi 30 funtów lub 30$ (około 650 tys. zł). ❑

EGS

24 bity = 16.7 mln. kolorów

[illegible]

Amerykańska firma GVP jest dobrze znana niemal od początku istnienia Amigi. Przez kilka lat działalności zdążyła wyrobić sobie doskonałą opinię. Jej wyroby cechuje solidność, nowoczesność i funkcjonalność. Tak jest i tym razem.

Instalacja karty jest bardzo prosta - polega na umieszczeniu jej w jednym z wolnych slotów Zorro 2/Zorro 3. Następnie należy wyjście RGB z Amigi połączyć z wejściem RGB karty (w opakowaniu znajduje się odpowiedni kabel), a do jej wyjścia trzeba podłączyć monitor. Po wykonaniu tych czynności można uruchomić komputer. Jeśli karta jest sprawna, to komputer powinien wystartować normalnie. Teraz z pierwszej dyskietki należy uruchomić program instalacyjny (na szczęście jest to dobrze znany Installer) i wykonywać jego polecenia. Następnie trzeba odpowiednio ustawić znaczną ilość różnych preferencji związanych z kartą, zależnie od posiadanego monitora i komputera.

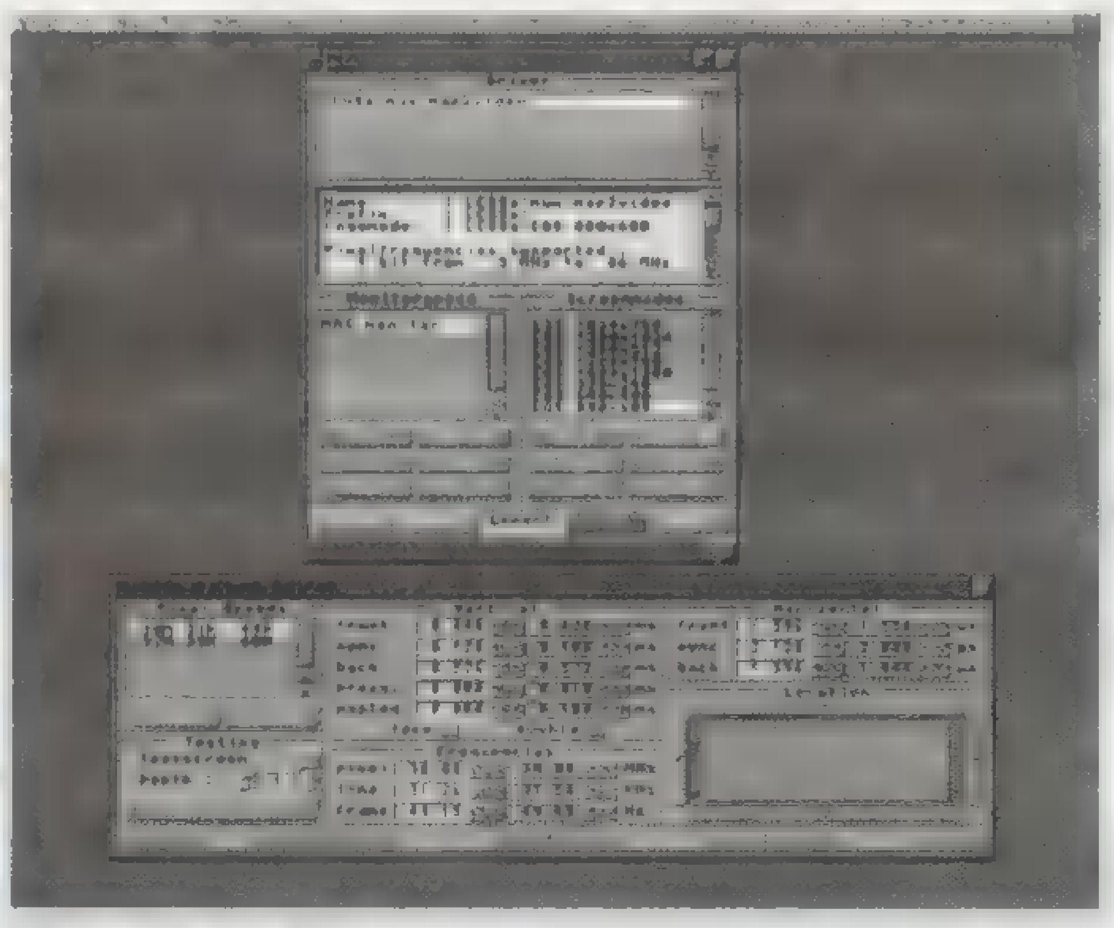
Przyjrzyjmy się bliżej parametrom karty oraz sposobowi w jaki komunikuje się ona z komputerem. Jak sama nazwa wskazuje, karta pracuje w standardzie EGS. Cóż to takiego jest EGS? Otóż jest to skrót od "Enhanced Graphic System", co można tłumaczyć w przybliżeniu jako "system graficzny o rozszerzonych możliwościach". Jednak EGS nie jest tylko zwykłym skrótem - jest to równocześnie nazwa nowego standardu kart graficznych do Amigi. Charakteryzuje się on niespotykaną dotychczas elastycznością. Programy w nim napisane pracują niezależnie od aktualnie posiadanego hardware'u. Oznacza to, iż dany program dla EGSa będzie działał na wszystkich kartach tego standardu, natomiast jego możliwości będą zależały tylko od posiadanego sprzętu. Jak wielkie ma to znaczenie dla producentów oprogramowania nie trzeba chyba wyjaśniać. Oprogramowanie systemu EGS jest praktycznie jednakowe dla wszystkich kart tego typu, niezależnie od producenta. Jedynym wyjątkiem są dwie biblioteki systemowe: EGS.library i EGSBlit.library, związane nierozłącznie z daną kartą. Tylko one "znają" parametry używanego aktualnie sprzętu. Wszystkie inne funkcje są niezależne od hardware'u i korzystają właśnie z tych bibliotek. Przejdźmy jednak do opisu samej karty. Przewidziana jest ona do Amig A2000/3000/4000. Sprzedawane są wersje z pamięcią 1 i 2 MB. Wersja 1 MB może służyć co najwyżej do zabawy. Różnica cenowa jest między nimi minimalna, tak że zdecydowanie odradzam zakup zubożonej wersji. Komunikacja z komputerem odbywa sie przez złącze Zorro 2 lub Zorro 3 (slot video pozostaje wolny!). Karta automatycznie rozpoznaje rodzaj złącza, jednak prawidłowa praca z Zorro 3 jest możliwa dopiero po wymianie układu Super-Buster na nowy (nowy chip musi mieć revision number 11). Dotyczy to zarówno A3000, jak i A4000. Jeżeli ktoś jeszcze nie ma tego układu, to należy wymusić pracę z Zorro 2 przestawiając odpowiedni jumper na karcie. W przeciwieństwie do innych kart graficznych (jak np. Retina), komunikacja z komputerem jest bardzo szybka, gdyż karta jest skonstruowana tak, że procesor widzi jej pamięć jak swoją własną. Ponadto istnieje możliwość otworzenia wielu ekranów wysokiej rozdzielczości o łącznej pamięci znacznie przekraczającej 2 MB. W takich przypadkach w pamięci karty znajduje się aktywny ekran, a inne są tymczasowo kopiowane do fastramu. Przełączanie takich ekranów odbywa się niemal tak szybko, jak w standardowej Amidze. Zależy to jednak od konfiguracji karty (Zorro 2, Zorro 3). W zwykłej A3000/25 MHz dane do i z karty przesyłane są z prędkością 3.5 MB/s (Zorro 2) lub 12 MB/s (Zorro 3). Jak z tego widać, nawet ekrany 24-bitowe o rozdzielczości 800x600 są przełączane bardzo szybko (ok. 0.25 sekundy dla Zorro2 i ok. 0.04 sekundy dla Zorro3). Bardzo duża szybkość karty jest możliwa dzięki jej własnemu superszybkiemu blitterowi, który kopiuje bloki pamięci z prędkością 56 milionów pikseli na sekundę. Karta wyposażona jest także w hardware'owy kursor o wymiarach 16x16, 32x32 lub 64x64 piksele (dwa razy dokładniejszy niż w A4000). Stosowanie pointera o maksymalnych wymiarach ma oczywiście sens tylko przy najwyższych rozdzielczościach, gdy zbyt mały kursor przestaje być widoczny.

Przykładowe maksymalne dostępne rozdzielczości wynoszą:

953 x 716 - 16777216 kolorów,
1168 x 876 - 65536 kolorów,
1600 x 1280 - 256 kolorów.

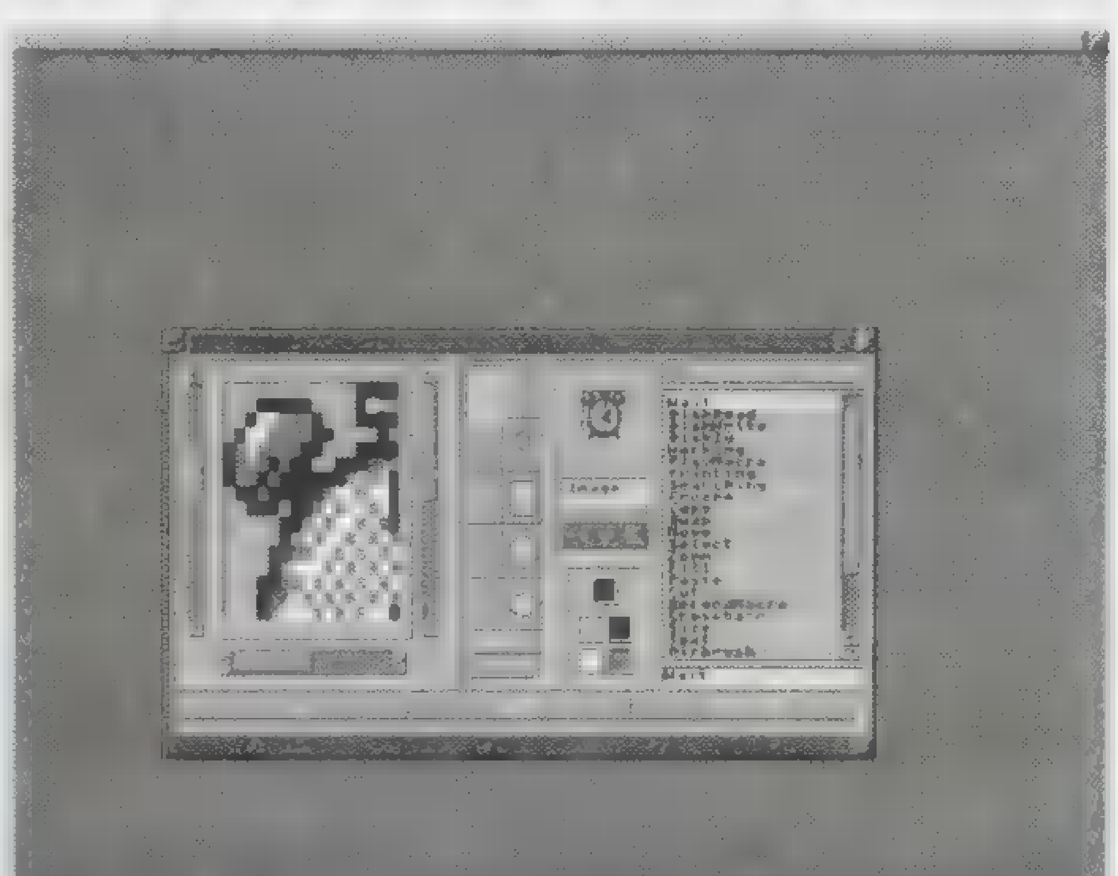
Zakresy synchronizacji:
Pozioma: max. 80 kHz,
Pionowa: max. 120 Hz.

Warto zaznaczyć, że wszystkie tryby graficzne są całkowicie programowalne. Możliwe jest ustawienie praktycznie dowolnej rozdzielczości, oczywiście pod warunkiem, że nie przekroczy się przy tym maksymalnej wielkości video ramu, tzn. 2 MB (dokładnie 2048000 bajtów). Programuje się nie tylko rozdzielczość, ale także częstotliwość odchylania poziomego i pionowego, a nawet takie szczegóły sygnału wideo jak front porch, back porch, synchro pulse, pre-equalization i post-equalization. Dzięki temu można do maksimum wykorzystać możliwości posiadanego monitora, a nawet skorygować jego niektóre wady! Ponadto karta jest przygotowana do mającego się niedługo pojawić systemu RTG (Retargetable Graphic). (Patrz Rysunek 1.)



Rysunek 1.

W systemie EGS okienka i gadżety są skonstruowane nieco inaczej niż w systemie opracyjnym Amigi. Wszystkie gadżety (a w szczególności slidery - belki do przesuwania zawartości okienka) automatycznie dopasowują się do aktualnych wymiarów ekranu. Jest to cecha której zdecydowanie brakuje amigowemu Intuition, szczególnie, gdy używany jest screen o rozdzielczości 640 x 512. W tej rozdzielczości dolny slider jest bardzo cienki i niewygodny w użyciu, podczas gdy slider boczny ma normalne wymiary. W EGSie oba slidery są identyczne, niezależnie od wybranego trybu graficznego. Pewną niedogodność stanowi brak możliwości "ściągania" ekranów za belkę tytułową, co wynika z zupełnie innej organizacji video ramu karty i Amigi (bitplane dla Amigi i chunky pixel dla EGSa). Dla screenów EGS można zdefiniować aż 31 różnych rodzajów pointera, które używane są przy odpowiednich operacjach (np. zapis na dysk, drukowanie, oczekiwanie, itd.)



Rysunek 2.

(Patrz Rysunek 2)

System okienek jest rozwiązany bardzo ciekawie. Jeśli w jakimś programie dostępne jest pull-down menu, to każdy punkt z tego menu może być "wyciągnięty" na zewnątrz, tworząc jak gdyby nowy gadżet dostępny w dowolnej chwili. Oczywiście taki gadżet może zostać ponownie zamieniony na zwykłe menu.

Wraz z kartą firma GVP dostarcza znakomity program graficzny EGS paint, określany przez firmę jako "real time true-color painting system". Ma on bardzo wiele funkcji niespotykanych dotychczas w innych programach, jest przy tym łatwy w obsłudze i rzeczywiście szybki. W 24 bitach pracuje on często szybciej niż DPaint w 16 kolorach! (Patrz Rysunek 3).

A jak ma się sprawa z Workbenchem i programami nie napisanymi specjalnie dla systemu EGS? Okazuje się, iż zależy to od konkretnej aplikacji. Oprogramowanie karty zawiera emulator Workbencha, dzięki któremu Workbench jest uruchamiany na dowolnym ekranie EGS. Dostępne nowe tryby graficzne wyświetlane są w preferencjach (screen mode). Jedyne ograniczenie to 256 kolorów. Wynika ono nie tyle samej karty, co systemu operacyjnego Amigi (Workbench może być co najwyżej 256 kolorowy). Każdy poprawnie napisany program korzystający z systemu będzie działał bez problemów. Z popularnych programów tego typu można tu wymienić PageStream, ArtExpression, QuarterBack, VirusChecker. Inna grupa programów to taka, która korzysta ze "screen data base" (potrzebny ekran może być wybrany przez użytkownika). W tym przypadku wskazane jest oczywiście wybranie ekranu EGS. Do takich programów należy między innymi Directory Opus, Term, TypeSmith, FinalCopy, FinalWriter, GPFax, BootX. Niestety, nie wszystkie takie programy działają bez zarzutu. Dotyczy to jednak tylko aplikacji nie używających systemu, lecz wykorzystujących różne "sztuczki" software'owe. Jest ich na szczęście coraz mniej. Trzecia grupa to programy otwierające własny, niedefiniowalny ekran (Imagine, ADPro, PowerPacker, AudioMaster). Dla nich karta SPECTRUM jest jakby "przezroczysta" - sygnał video z Amigi przechodzi przez nią nie zmieniony. Takie rozwiązanie ma bardzo istotną zaletę: do wszystkiego potrzebny jest tylko jeden monitor.

W jednym komputerze można zamontować nawet kilka kart EGS, a każda może mieć podłączony własny monitor. Praca wielomonitorowa jest bardzo prosta - kursor po osiągnięciu brzegu jednego monitora przeskakuje do następnego.

Oprogramowanie pracujące na ekranach EGS działa zawsze o wiele szybciej niż na zwykłych amigowych. Ostatnio pojawia się coraz więcej programów dla tego standardu. Ciekawym przykładem może być programik do odgrywania... PCtowych 8-bitowych animacji w rozdzielczości 320 x 200. Szybkość takich animacji na karcie SPECTRUM przekracza 150 klatek na sekundę (!), a może ich być kilka na jednym ekranie!

Swoimi osiągami karta SPECTRUM znacznie przekracza możliwości grafiki dostępnej w Amidze 4000. Niestety, nie emuluje ona nowych układów AA, więc nie będą na niej działać praktycznie żadne "niedosowe" gry napisane specjalnie dla A1200/A4000. Nie nadaje się więc ona dla namiętnych graczy. Natomiast z czystym sumieniem można ją polecić wszystkim zajmującym się na codzień grafiką komputerową oraz DTP. Cena karty EGS 28/24 SPECTRUM (2 MB) wynosi około 900 DM i jest jak najbardziej adekwatna do jej możliwości. ❑



Rysunek 3.

HP LaserJet 4L

Tomasz Flanc

Oficjalna światowa premiera tej drukarki odbyła się w maju tego roku, choć już kilka miesięcy wcześniej wiadomo było, że firma Hewlett-Packard szykuje coś specjalnego.

Tym razem Hewlett-Packard postanowił zdobyć ogromny rynek przeciętnego klienta o niezbyt zasobnym portfelu, o wygórowanych wymaganiach, dla którego nieosiągalną jest drukarka klasy HP-4, a osiągnięcia "atramentówek" wydają się mu mizerne. W zamyśle Hewlett-Packarda LaserJet 4L zaprojektowano właśnie jako najtańszą drukarkę laserową dla indywidualnego użytkownika w domu lub małej firmie, mającą zastąpić popularną drukarkę HP LaserJet IIP. Niską cenę osiągnięto dzięki uproszczonej konstrukcji oraz ograniczeniu możliwości rozbudowy. Nie odbyło się to jednak kosztem obniżenia jakości, gdyż jednocześnie wprowadzono kilka rewolucyjnych zmian, do których należy lepsze zarządzanie pamięcią oraz likwidacja panelu sterującego.

Nominalna szybkość druku wynosi cztery strony na minutę. Nie jest to może oszałamiająca ilość w porównaniu z innymi maszynami, ale do domowych zastosowań w zupełności wystarczająca. Druk może odbywać się na papierze, folii, nalepkach, kopertach w formatach do 216x356 mm. Papier podawany jest z kasety mieszczącej około 100 arkuszy lub pojedynczo - ręcznie. Podczas pobierania z kasety i odkładania na wierzch drukarki, papier przechodzi drogę w kształcie litery S. Przy ręcznym podawaniu i odkładaniu za drukarką, droga papieru jest niemal prostoliniowa.

Druk odbywa się ze standardową rozdzielczością 300 dpi, przy czym jego jakość podnosi wbudowany 3-poziomowy mechanizm wygładzania krzywych RET (Resolution Enhancement Technology). Na podwyższenie jakości druku wpływ ma również specjalny, drobnoziarnisty toner wypełniający mniejszą niż dotychczas kasetę. Mniejsza kaseta nie oznacza jednak zmniejszenia wydajności drukarki. Możliwe jest bowiem drukowanie w trybie ekonomicznym, dającym o połowę mniejsze zużycie toneru przy nieznacznej utracie jakości wydruku. Jest to bardzo praktyczne, gdy wykonuje się wiele wydruków próbnych.

Nowy algorytm zarządzania pamięcią i kompresji przesyłanych danych MET (Memory Enhancement Technology) powoduje, że w większości przypadków wystarcza standardowa pamięć 1MB, którą w razie potrzeby można podwoić. Dzięki tej technice rzadziej pojawia się, wynikający ze zbyt małej szybkości obróbki danych, tzw. Overrun Error. Można bez przeszkód wydrukować w najwyższej rozdzielczości całą stronę A4.

Komputer z drukarką komunikuje się przez znany z HP-LJ4 dwukierunkowy interfejs Bi-Tronics, sprzętowo i programowo zgodny z Centronicsem, ale od niego szybszy i umożliwiający przekazywanie informacji z drukarki do komputera. Właśnie dzięki możliwości dwustronnej komunikacji, panel sterujący drukarki ograniczono do czterech diod sygnalizacyjnych i jednego klawisza. Jednym z zadań panelu sterującego było dotychczas wyświetlanie komunikatów. Odbywało się to za pośrednictwem małego wyświetlacza ciekłokrystalicznego, którego w drukarce LJ-4L zastępuje specjalne oprogramowanie. I tu pojawia się pierwszy problem: oprogramowanie dołączone do drukarki opracowane zostało dla popularnych PC-tów. Co w takim razie ma zrobić posiadacz Amigi, Atari czy innego komputera? Czyżby Hewlett-Packard przekreślił z listy swoich potencjalnych klientów użytkowników tych komputerów? Zlikwidowano również wyłącznik sieciowy, drukarka jest stale pod napięciem i zależnie od poziomu aktywności znajduje się w jednym z trzech stanów - drukowania, oczekiwania lub uśpienia. W stan uśpienia automatycznie przechodzi po 15 minutach bezczynności, pobiera wówczas tylko ok. 5 W (tyle co telewizor w trybie STAND BY). Stan ten sygnalizuje zgaśnięcie lampki kontrolnej, a rozbudzenie następuje w chwili wysłania danych z komputera lub dotknięcia jedynego klawisza panelu.

W drukarce zastosowany jest najnowszy mechanizm Canona - mała kaseta z drobnoziarnistym tonerem. Jednostką obliczeniową jest dobrze nam znany procesor Motorola 68000 (z zegarem 16MHz).

Sterowanie parametrami drukarki odbywa się całkowicie zdalnie z komputera za pomocą rezydentnego programu. Za jego pośrednictwem można ustawić krój liter, włączyć wspomniany wcześniej tryb wygładzania krzywych czy zainicjować tryb oszczędności toneru. Niestety brak oprogramowania, przekreśla tą możliwość dla posiadaczy wielu komputerów (w tym także Amigi). Jedynym ratunkiem dla nich jest sięgnięcie po specjalne narzędzia służące do kontroli wydruku. Jak się dowiedzieliśmy program Studio firmy Makrosystem posiada drivery do tej drukarki. Wypada, więc z nich skorzystać.

Konkludując powyższe fakty można stwierdzić, że drukarka zasługuje na ocenę bardzo dobrą.

HP LaserJet 4L
Producent: Hewlett-Packard
Wymiary: 362x353x164
Waga: 7 kg
Szybkość druku: 4 strony/min.
Cena: ok. 20 mln zł. (łącznie z tonerem)

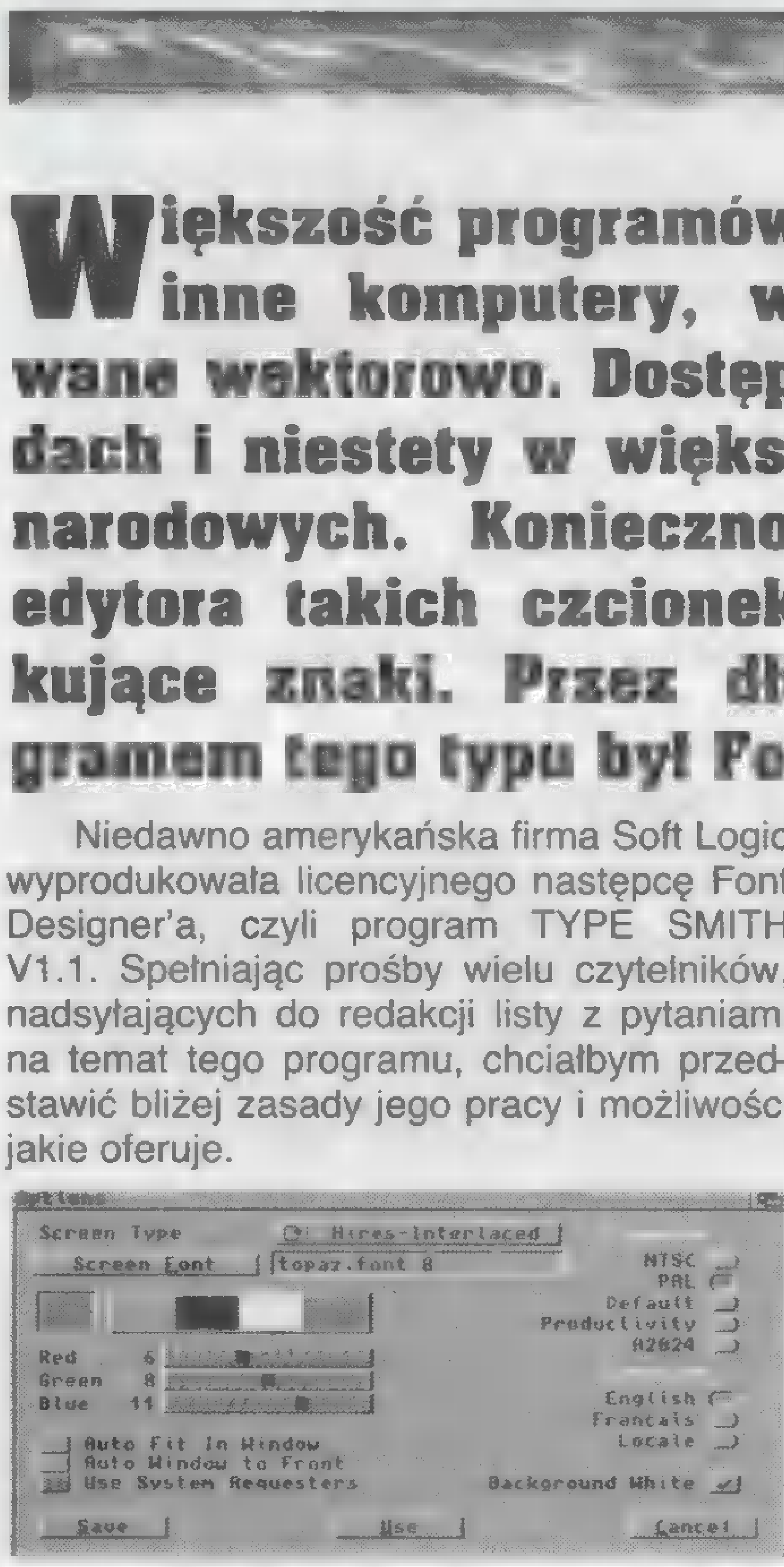


TYPE SMITH

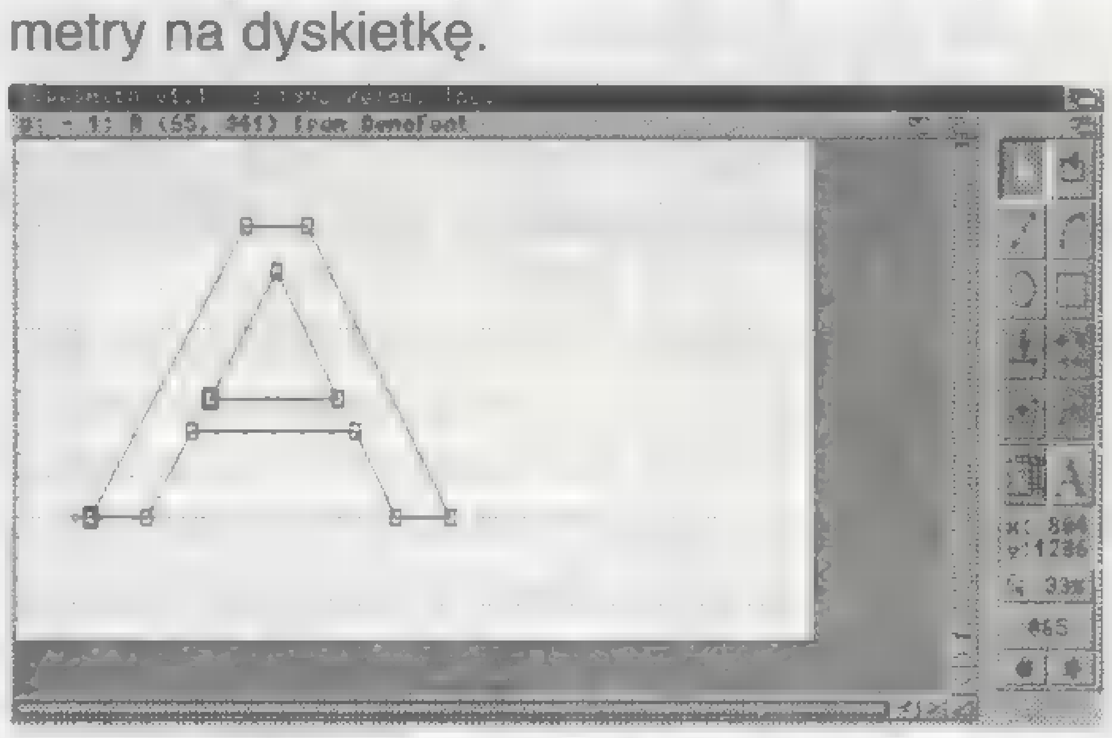
Tomasz Hrycuniak

Większość programów DTP na AMIGĘ, a także na inne komputery, wykorzystuje fonty definiowane wektorowo. Dostępne są one w kilku standardach i niestety w większości bez polskich znaków narodowych. Koniecznością jest więc używanie edytora takich czcionek, aby uzupełnić je o brakujące znaki. Przez długi czas popularnym programem tego typu był Font Designer.

Niedawno amerykańska firma Soft Logic wyprodukowała licencyjnego następcę Font Designer'a, czyli program TYPE SMITH V1.1. Spełniając prośby wielu czytelników, nadsyłających do redakcji listy z pytaniami na temat tego programu, chciałbym przedstawić bliżej zasady jego pracy i możliwości jakie oferuje.

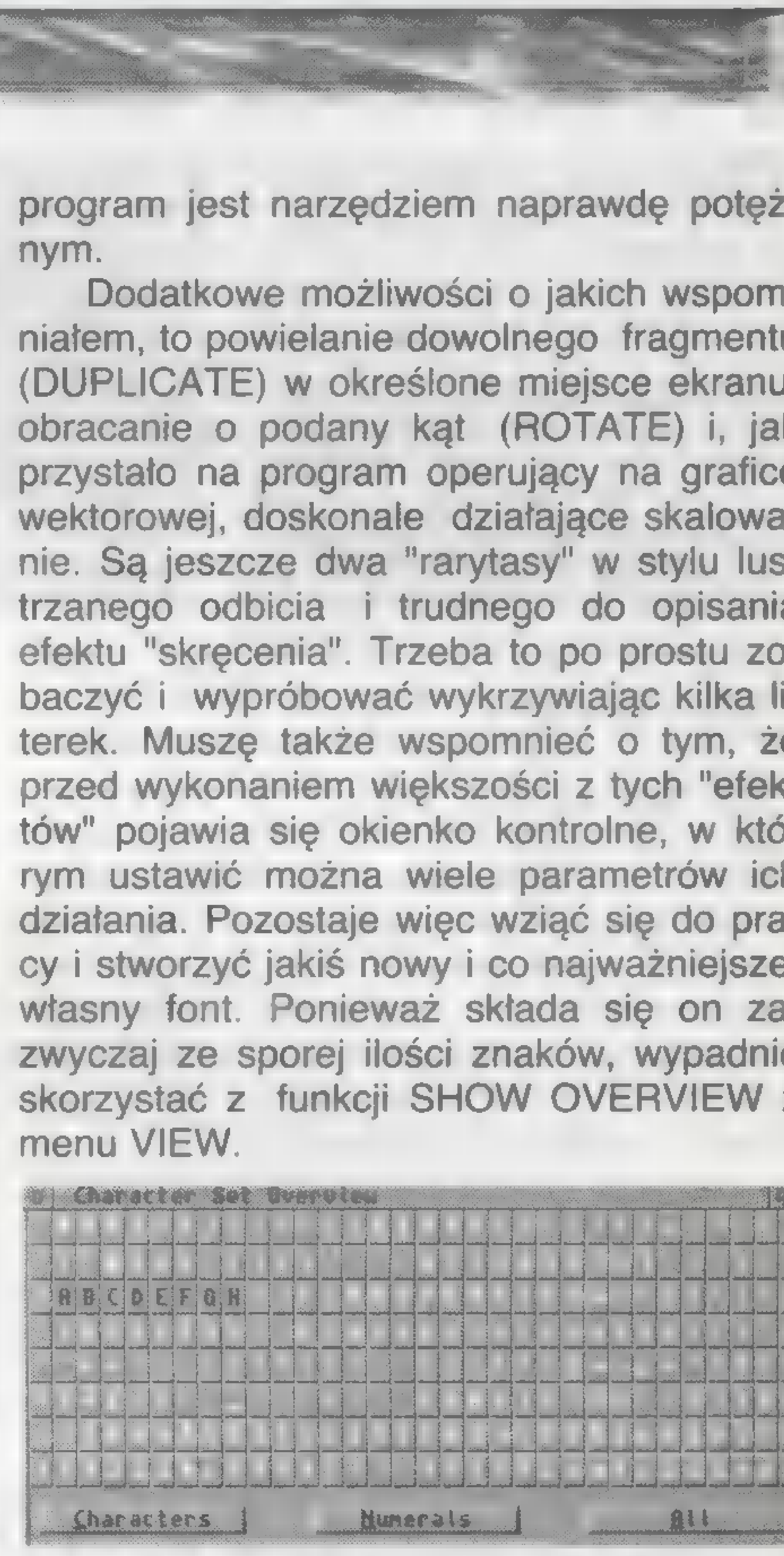


Pracę z programem Type Smith najlepiej rozpocząć od ustawienia preferencji tak, aby dostosować go do swoich potrzeb. Szereg opcji do tego przeznaczonych znajdziemy w menu SETTINGS. Na samym początku znajdują się funkcje związane z "siatką" pomocniczą. Oprócz jej włączenia i wyłączenia, możemy ustawić wielkość "oczek" oraz uruchomić funkcję "skokowego" poruszania się po nich kursora. Poza siatką, w oknie roboczym można umieścić także linie pomocnicze. Są one bardzo potrzebne przy rysowaniu liter "od zera", gdyż pozwalają na utrzymanie właściwej szerokości i wysokości znaków. Uruchomienie funkcji OPTIONS wywołuje na ekran okno kontrolne zawierające wiele różnorodnych opcji. Poczynając od ustawienia kolorów ekranu, przez wybranie trybu wyświetlania i fontu używanego w programie, na języku, w którym mają ukazywać się komunikaty kończąc. Korzystając z menu SETTINGS możemy kontrolować współpracę programu z AREXXem, ustalić ścieżki dostępu do używanych katalogów i oczywiście zgrać wszystkie te ustawione parametry na dyskietkę.



Po wybraniu opcji NEW z menu PROJECT, ujrzymy na ekranie okno robocze i możemy przystąpić do tworzenia nowej czcionki. Podstawowe "narzędzia" do tej pracy umieszczone są w postaci ikon na listwie widocznej w prawej części ekranu. W tym miejscu zaznaczyć muszę, że operacja "rysowania" przebiega tu podobnie jak w programach do tworzenia grafiki wektorowej (np. ProDraw). Osoby, które nie miały z nimi kontaktu mogą mieć trudności z poprawną obsługą tych funkcji. Szczególną uwagę trzeba zwrócić na rysowanie linii krzywych. Nie jest to może specjalnie trudne, ale ma swoją specyfikę i wymaga pewnego treningu. Mamy za to do dyspozycji szereg funkcji pomocniczych, które w wielu przypadkach bardzo ułatwiają pracę. Stosując opcje dostępne w menu EDIT możemy swobodnie wycinać, kopiować do pamięci i przywoływać ponownie, dowolne fragmenty rysunku, a także skorzystać z operacji przywrócenia poprzedniego stanu w przypadku popełnienia błędu (UNDO). Jeśli wszystko to wzbogacimy jeszcze dodatkowymi możliwościami z menu PATCH, to okaże się, że w dziedzinie rysowania opisywany program jest narzędziem naprawdę potężnym.

Dodatkowe możliwości o jakich wspomniałem, to powielanie dowolnego fragmentu (DUPLICATE) w określone miejsce ekranu, obracanie o podany kąt (ROTATE) i, jak przystało na program operujący na grafice wektorowej, doskonale działające skalowanie. Są jeszcze dwa "rarytasy" w stylu lustrzanego odbicia i trudnego do opisania efektu "skręcenia". Trzeba to po prostu zobaczyć i wypróbować wykrzywiając kilka literek. Muszę także wspomnieć o tym, że przed wykonaniem większości z tych "efektów" pojawia się okienko kontrolne, w którym ustawić można wiele parametrów ich działania. Pozostaje więc wziąć się do pracy i stworzyć jakiś nowy i co najważniejsze, własny font. Ponieważ składa się on zazwyczaj ze sporej ilości znaków, wypadnie skorzystać z funkcji SHOW OVERVIEW z menu VIEW.



Wyświetla ona na ekranie okno z całym "alfebetem". Podwójne kliknięcie na wybranym znaku, przenosi nas do okna roboczego pozwalającego na jego edycję. Do poszczególnych znaków można dostać się także używając strzałek widocznych w prawym dolnym rogu ekranu głównego lub po prostu wciskając właściwe klawisze. Jeśli jestem już przy menu VIEW to przypomnę, że znajdują się tam opcje ZOOM IN i ZOOM OUT, czyli powiększanie i pomniejszanie rysowanego znaku oraz FIT IN WINDOW przywracającą takie jego wymiary, aby cały był widoczny na ekranie. Tworzoną właśnie czcionkę można wypróbować przy pomocy opcji SHOW TYPE PREVIEW wyświetlającej dodatkowe okno, w którym wpisujemy kilka wyrazów i sprawdzamy jak nasz font będzie się sprawował w praktyce.

Jest jeszcze jedna możliwość ułatwienia sobie pracy. Skorzystanie z tzw. podkładu. Rysunek jest widoczny i używając standardowych funkcji Type Smitha możemy rysować "po jego krawędziach". Jest to o wiele łat

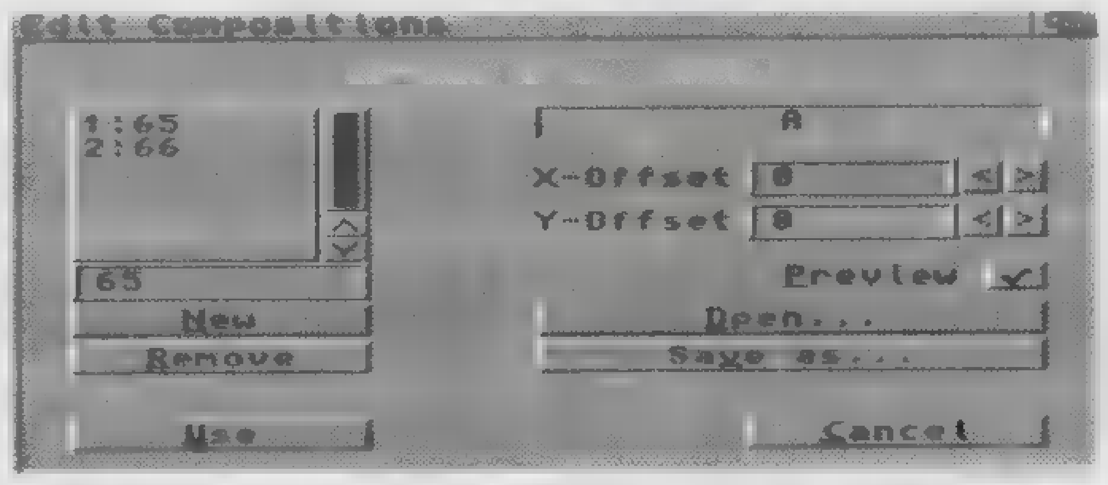
wiejsze niż rysowanie z pamięci. Podkład oczywiście można usunąć, otrzymując prawie gotową literę. Wystarczy dokonać drobnych poprawek w celu wyeliminowania niedokładności. Jako "podkład" może posłużyć nam dowolny rysunek w formacie IFF (np. z programu Deluxe Paint) lub standardowa czcionka bitmapowa. Wzorce importujemy za pomacą funkcji IMPORT z menu PROJECT.

Funkcja IMPORT poza wprowadzaniem "podkładu" pozwala na wczytanie grafiki wektorowej w formacie IFF DR2D, przyjętym na AMIDZE jako jeden ze standardów i dostępnym między innymi w programie "Art Expression". Ma to duże znaczenie, ponieważ taka grafika może zostać przekształcona bezpośrednio na znak w tworzonej przez nas czcionce. Importem można posłużyć się także w wypadku innych standardów czcionek, gdyż podstawowym formatem dla Type Smitha, podobnie jak PageStreama, jest format własny firmy Soft Logic (DMF fint) i jest on obsługiwany przez opcje OPEN i SAVE. Dzięki opcji IMPORT, mamy dostęp do bogatych bibliotek fontów w standardach Compugraphic Intelifont (Agfa Compugraphic) oraz PostScript Type 1 (Adobe Type 1). W połączeniu z podobnie szerokim, a nawet poszerzonym o PostScript Type 3 (Adobe Type 3) i grafikę wektorową IFF DR2D eksportem pozwala to stosować opisywany program jako konwerter fontów, z całkiem dobrymi wynikami. Jako ciekawostkę mogę podać, że w oknie pojawiającym się przy eksporcie w standardzie Compugraphic Intelifont, wśród wielu innych, jest także opcja pozwalająca na przygotowanie czcionek dla bądź co bądź konkurencyjnego do PageStream'a programu ProPage.

Jeśli jest mowa o wczytywaniu i edycji gotowych fontów, to czas najwyższy na kilka zdań w temacie spolszczenia czcionek, czyli uzupełnienia ich o ą, ę, ó, itd. Najprostszym choć, jak się za chwilę okaże, wcale nie najlepszym rozwiązaniem jest wykasowanie znaków znajdujących się pod kodami przeznaczonymi dla polskich liter i narysowanie w ich miejscu potrzebnych nam znaków. Trzeba przy tym jednak zachować jeden ze stosowanych w naszym kraju standardów. Kody naszych znaków narodowych można znaleźć w wielu publikacjach poświęconych AMIDZE, a i w AMIGOWCU były swego czasu drukowane przy okazji odpowiedzi na list jednego z czytelników.

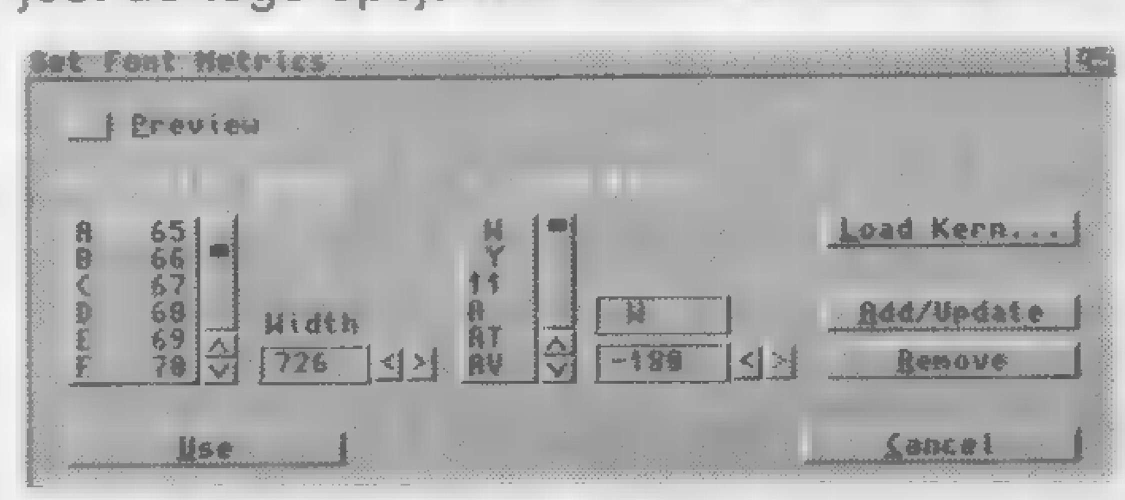
Zamiast żmudnego rysowania nowych liter, użyć można funkcji COMPOSITIONS z menu EDIT. Pozwala ona na nałożenie na siebie kilku znaków będących już w zestawie, w celu uzyskania nowego znaku i umieszczenie go pod określonym kodem ASCII. Operację taką przedstawię na przykładzie litery "ż". Pracę należy rozpocząć od wywołania na ekran znaku o kodzie właściwym dla litery "ż" (zależnie od przyjętego standardu). Następnie usuwamy znak stający na miejscu gdzie chcemy ustawić własną literę i uruchamiamy funkcję COMPOSITIONS.

Pojawi się nowe okno, w którym trzeba ustawić kilka parametrów. Najważniejsza jest ramka widoczna w lewej części okna.

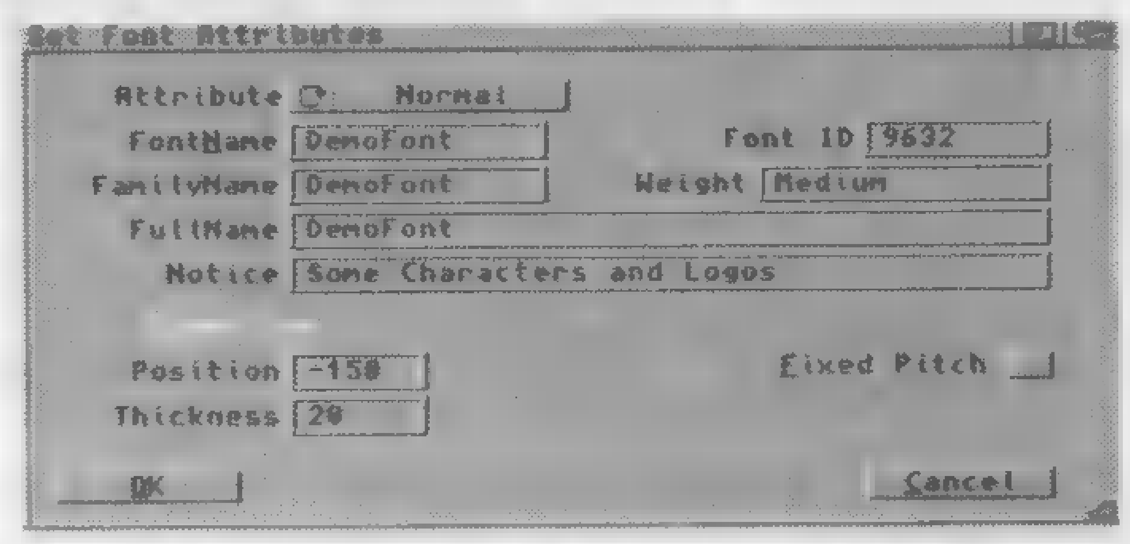


Korzystając z opcji NEW wpisać należy kody znaków jakie chcemy na siebie nałożyć i umieścić w aktualnie widocznym oknie roboczym. W naszym przypadku będą to kody litery "z" i "." (kropki). W ich ustaleniu pomocna jest funkca SHOW OVERVIEW. W następnej kolejności trzeba ustawić odpowiednie położenie naszych znaków względem siebie tak, aby tworzyły literę "ż". Pozostaje więc wpisanie odpowiednich liczb przy X-Offset i Y-Offset. Prościej jest jednak włączyć opcję PREVIEW i przesunąć myszką okno EDIT COMPOSITIONS tak, by widzieć dokładnie oba nakładane znaki. Klikając na strzałki przy X-Offset i Y-Offset można przesuwać jeden znak względem drugiego i w ten sposób ustalić ich odpowiednie położenie. Podobnie należy postępować w przypadku innych liter. Oprócz "kropki" stosować można także inne znaki np. "," (przecinek). W razie potrzeby istnieje możliwość odpowiedniej modyfikacji, czyli wykonania: obrótu, lustrzanego odbicia lub nieznacznego wykrzywienia. Po nabraniu odrobiny wprawy "spolszczanie" czcionki idzie dosyć szybko i z dobrym efektem. Trzeba jednak uważać przy fontach "stylizowanych". Przy takich "wygibaśnych" literkach nasz "przecinek" może nie wyglądać zbyt estetycznie i "ą" czy "ę" będą prezentować się raczej dziwnie. Pozostanie wtedy wykazać trochę inwencji, chwycić za myszkę i wykonać własnoręcznie "haczyk" stosowny do stylu obrabianej czcionki. Stworzenie wszystkich polskich literek nie jest niestety końcem naszej pracy. Przed zgraniem poprawionego fontu trzeba będzie wykonać jeszcze jedną, bardzo ważną operację.

Mam na myśli ustalenie odpowiednich odległości między literami. Tak się składa, że litery mają różne kształty i odległości między nimi nie mogą być identyczne. Korekty takiej możemy dokonać w edycji tabeli odległości między znakami. Przeznaczona jest do tego opcja METRICS z menu EDIT.



Po jej uruchomieniu ukazuje się okno zatytułowane SET FONT METRICS. Zawiera ono pola ze spisem wszystkich znaków naszego fontu i zdefiniowanych już par liter. Uwaga! Każdy font powinien posiadać osobny plik zawierający wyżej wymienione informacje. Podczas edycji nie wolno zapomnieć o wczytaniu tego pliku. Dokonujemy tego przez OPEN METRICS z menu PROJECT w przypadku fontu w standardzie firmy Soft Logik (DMF) lub IMPORT AFM POST SCRIPT METRICS w przypadku fontu Post Script'owego (Adobe). Jeżeli stworzyliśmy nowy font czeka nas sporo pracy, ale gdy dokonaliśmy tylko "spolszczenia" wypadnie poprawić tylko odległości nowo wprowadzonych znaków. Jest to operacja dosyć trudna, gdyż trzeba zachować parametry przyjęte przy pozostałych literach w czcionce. W tym miejscu rada: najłatwiej jest wzorować się na literach podobnych do naszych nowo wprowadzonych znaków. Np. w przypadku litery "ą" na literze "a", "ó" na "o" itd. Trzeba po prostu spisać sobie definicje litery "a" i uzyskaną wartość przepisać do definicji dotyczącej litery "ą", itd. Ustawienie takich samych parametrów np. dla pary "ą" i "t" jak dla pary "a"i "t" (kerning) zagwarantuje poprawny wygląd tekstu pisanego naszą poprawioną czcionką. W razie wątpliwości, efekty naszej pracy trzeba sprawdzić używając funkcji "PREVIEV" w oknie SET FONT METRICS (przegląd pary znaków) lub uruchomić wspomnianą już funkcję SHOW TYPE PREVIEW z menu EDIT i napisać kilka wyrazów nowo wprowadzonymi literami. Jeżeli wszystko będzie w porządku, można przystąpić do ostatniej, choć niekoniecznej operacji, czyli zmian nazwy fontu. Służy do tego funkcja FONT ATRIBUTES z menu EDIT.



Wyświetla ona okno, w którym można zmienić nazwę czcionki lub dopisać końcówkę "PL" potwierdzającą wykonanie naszej pracy. Teraz jest już naprawdę koniec i można wszystko nagrać na dyskietkę w wybranym standardzie. Starałem się w miarę dokładnie przedstawić metodę spolszczenia fontów, gdyż zdaję sobie sprawę, że ze względu na możliwość wczytywania i konwersji czcionek we wszystkich liczących się standardach będzie to podstawowe zastosowanie opisywanego programu.

Na tym kończę moje wywody na temat Type Smitha i myślę, że po wypróbowaniu przedstawionych tu możliwości, wielu czytelników AMIGOWCA dołączy do grona zagorzałych zwolenników tego programu. Jest on naprawdę doskonały i zwyczajnie nie ma do czego się przyczepić. Muszę jednak ostrzec wszystkich przed używaniem programu przy 1MB pamięci. W takiej konfiguracji co prawda się uruchamia, ale już z samym wczytaniem, nie mówiąc o edycji większego fontu, mogą być problemy. Zatem 2MB wydają się koniecznością. Mimo to uważam, że właśnie dzięki temu programowi za jakiś czas zniknie problem czcionek w DTP na AMIDZE, irytujący dotąd wielu interesujących się tym tematem użytkowników. W drodze jest już nowa wersja programu TYPE SMITH wersja 2.0, która według moich informacji będzie wielką rewelacją. ❑

REAL 3D V.2

Krzysztof Trojanowski

Dopiero niedawno ukazała się długo oczekiwana wersja 2.0 programu Real 3D zapowiadana już na początku roku. To, co można było zobaczyć na targach lub odczytać z materiałów reklamowych, zapowiadało się bardzo obiecująco. W artykule tym podejmiemy próbę przekonania się, czy dostarczony produkt sprostał bardzo wygórowanym oczekiwaniom szerokiej rzeszy użytkowników programów ray-tracing.

Za cenę ok. 1000 DM mamy już nasz wymarzony program. W skład zestawu wchodzą: cztery dyskietki (zawierają one wersję programu dla procesora 68020/30 z koprocesorem (68881/82) oraz specjalną wersję dla procesora 68040, ponadto programy do generowania animacji, przeróżne konwertery obiektów, jak również całą masę przykładowych obiektów i projektów.), koperta, segregator formatu A5 i 450 stron grubego podręcznika. Podręcznik jest napisany zrozumiale i nie ma problemów z wyszukaniem potrzebnych informacji. Podręcznik zawiera sześć części:

WSTĘP - Wprowadza do programu i pomaga zacząć pracę.

PRZEWODNIK - Zaznajamia ze wszystkimi funkcjami w łatwych krokach z wieloma przykładami. W katalogu "Examples" są także pliki, które w tym pomagają.

OPIS FUNKCJI - Zawiera opis fukcji menu i gadżetów.

SŁOWNIK - REAL używa wielu specjalistycznych określeń, które są tu zdefiniowane.

DODATKI - Diagramy z ikonami, opis przyporządkowań klawiszy, używanie dołączonego oprogramowania.

INDEKS - Pełny indeks.

W rozdziale 4. znajduje się specjalny projekt do wczytania i instrukcje, którymi musi posłużyć się każdy użytkownik, aby trochę poeksperymentować i zasmakować mocy REALA 3D V.2.

Hardware

Osobną sprawę stanowi koperta. Jej otwarcie ujawnia dongle'a, który ma chronić program przed pirackim kopiowaniem. Dongle jest przewidziany dla portu joystika Amigi, toteż posiadacz programów "Scala Multimedia" lub "Dyna CAD", zabezpieczonych w ten sam sposób, nie uniknie uciążliwego przekładania tego zabezpieczenia podczas równoległej pracy z tymi programami. Kto nie chce podejmować ryzyka zamiany dongle'a przy włączonej Amidze musi obyć się bez multitaskingu. Realsoft zapowiedział już wypuszczenie dongle dla portu równoległego.

Zainstalowanie programu następuje wprawdzie bez commodorowskiego instalatora, lecz nie nastręcza to żadnych problemów.

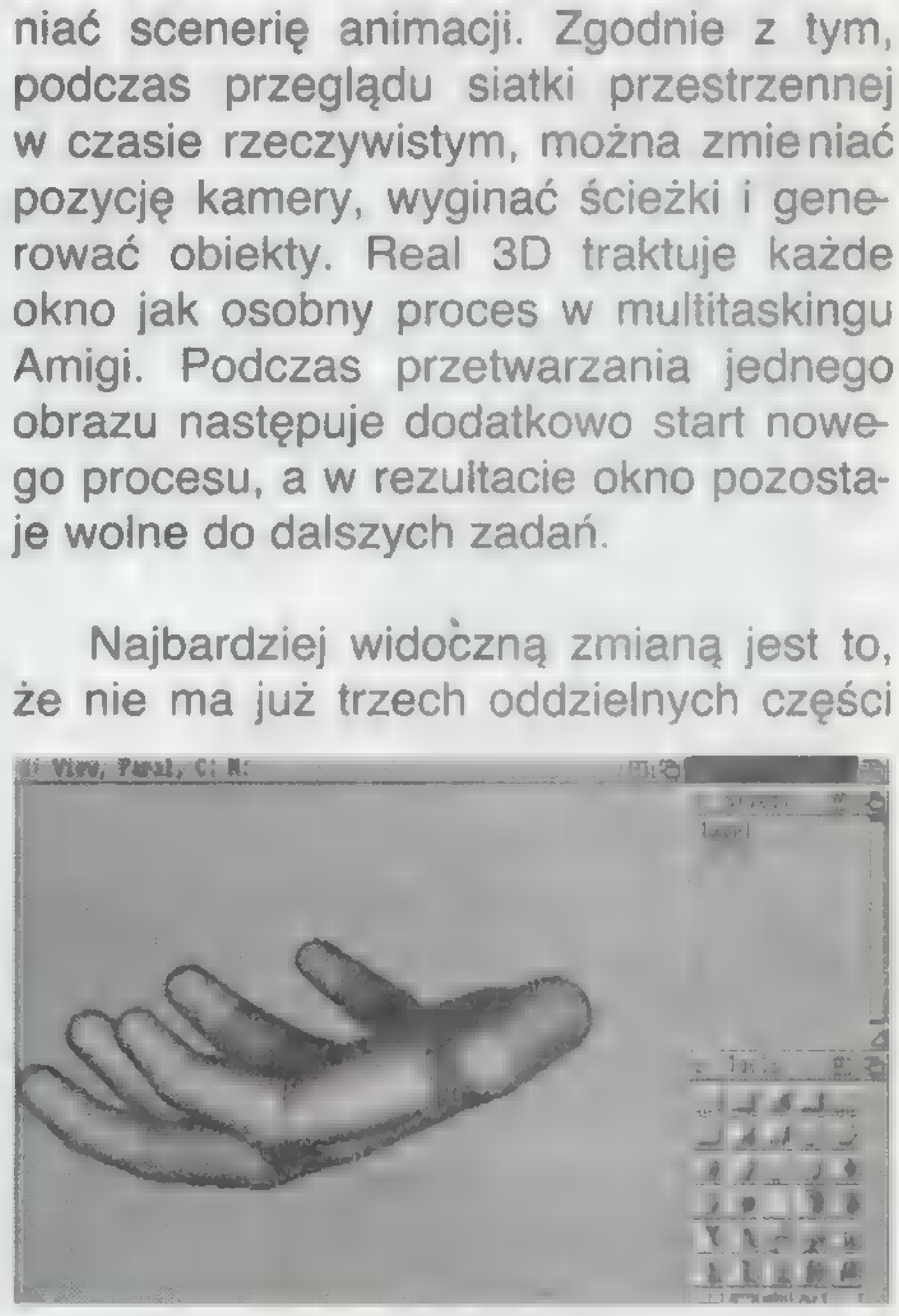
REAL 3D V.2 ma następujące wymagania sprzętowe:

- co najmniej 3MB RAM,
- twardy dysk z co najmniej 5MB wolnej pamięci na program,
- procesor MC68020 lub wyższy (np 68030/68040),
- koprocesor matematyczny.

Dla profesjonalnych użytkowników zaleca się konfigurację z co najmniej 5MB RAM i 10MB wolnej pamięci na twardym dysku zarezerwowanej dla środowiska REALA. Zoptymalizowana wersja programu dla procesora 68040 jest według testów ośmiokrotnie szybsza od wersji dla 68030 pracującego na tej samej częstotliwości zegara.

Dzięki asynchronicznej pracy programu oraz zastosowaniu najnowocześniejszych technik programowania obiektowego, REAL w pełni wykorzystuje wielozadaniowość systemu operacyjnego. Osiąga to przez maksymalną wygodę i wydajność modelowania. Użytkownik w żadnym przypadku nie musi czekać na zakończenie działania pewnych operacji, może kontynuować pracę dalej, podczas gdy inne zadania są wykonywane w tle. Liczba otwartych okien jest ograniczona tylko przez pamięć, którą mamy do dyspozycji. Pomysł ten nosi nazwę "Zero-Wait-State-Design", to znaczy, że podczas renderowania w jednym oknie możemy jeszcze dodawać nowe obiekty i zmieniać scenerię animacji. Zgodnie z tym, podczas przeglądu siatki przestrzennej w czasie rzeczywistym, można zmieniać pozycję kamery, wyginać ścieżki i generować obiekty. Real 3D traktuje każde okno jak osobny proces w multitaskingu Amigi. Podczas przetwarzania jednego obrazu następuje dodatkowo start nowego procesu, a w rezultacie okno pozostaje wolne do dalszych zadań.

Najbardziej widoczną zmianą jest to, że nie ma już trzech oddzielnych części



programu, jak miało to miejsce w "Imagine", "Reflections" lub w "Realu 1.x". W REALU 3D V.2 tworzenie obiektu, modyfikacja, animacja i renderowanie mogą być wywoływane z jednego ekranu. Oczywiście jest możliwe, a często konieczne, używanie większej ilości ekranów, jednak sposób ich użycia jest konfigurowalny przez użytkownika. Jest to realizowane przez umożliwienie użytkownikowi otwarcia potrzebnego ekranu i otwieranie więcej niż jednego okna na tym ekranie. Okna te mają określone funkcje w zależności od ich typu, lecz każde z nich może być skonfigurowane zgodnie z potrzebą obsługującego. Możliwe jest także deklarowanie ekranów jako ekranów publicznych (od wersji OS 2.0 - public screen). Oznacza to, że niektóre programy kompatybilne z OS 2.0 mogą być dołączone do REALA 3D. Również REAL może zostać dołączony do innych programów wykorzystujących tę cechę OS 2.0.

Konfiguracja ekranów i okien może zostać zapisana jako środowisko, a później wczytana w dowolnej chwili. Konfigurowalne środkowisko pozwala na tworzenie gadżetów definiowanych przez operatora i używanie ich jako jednoklawiszowych makrodefinicji rozszerzających interfejs użytkownika. Takie rozwiązanie czyni pracę na Realu bardzo komfortową.

Program wykorzystuje oczywiście AA-Chips Amigi 1200 i 4000. Do programu dołączane są również drivery do popularnych kart 24 bitowych (Impact Vision, Opal Vision, Retina), jak również gotowe struktury edytorów. Przy samych tylko oknach projekcyjnych można wybierać między czterema wariantami: "Normal", "Double Buffered" do animacji Previews, "Borderless" do wypełnionego obrazu i "Superbitmaps" do potężnych okien edytorskich. Oprócz tego istnieją jeszcze: edytor materiałów, okno z paletą, Animation/Control-Window (podobny do panelu sterowania w video), Measuring-Window (wyświetlający dane koordynacyjne i wymiary przy konstrukcjach pedantycznie dokładnych), okno RPL (wykorzystywane do bezpośredniego podawania poleceń RPL) oraz okno Screen/Control.

Metody renderingu i rewolucyjne innowacje

Na szczególną uwagę zasługuje nowa konstrukcja okien perspektywicznych. Mogą one pokazywać tworzoną scenę ze wszystkich stron, pod każdym kątem widzenia. Modelowana siatka, dane koordynacyjne i sieć przestrzenna w płaszczyznach XY pomagają przy orientowaniu się w przestrzeni 3D. Przy pomocy "Drawing Setting" możemy zmieniać różne parametry wyświetlania projektu np.: dokładność ukazania krzywizn, widoczność tekstur i punktów kontrolnych, realizację budowy ekranu i wiele innych. "Render-Setting" określa Ray-Trancing-Modus dla każdego okna projekcyjnego, rodzaj ditheringu, rozdzielczość (max. do 32 000 x 32 000), dateformat (IFF, Targa, Targa Plus, Alpha-Channel, Windows BMP), ostrość głębi, odzwierciedlenie półobrazu, przebiegi, obraz tła, wykorzystanie pamięci, parametry wyświetlania i renderingu. Program oferuje następujące metody renderowania:

- Draft - źródło światła z punktu obserwacji, bez materiałów, cieni,
- B-Splines, Anti-Aliasing,
- Environment - razem z materiałami,
- B-Splines, refleksami z Einvironment-Map i z Anti-Aliasingiem odpowiada Scanline-Render,
- Lampless - wraz z refleksami i jednym źródłem światła,
- Shadowless - dysponuje dowolnie wieloma własnymi źródłami światła,
- Normal - renderuje z cieniami.

Można używać nieograniczonej liczby źródeł światła o dowolnym kolorze i jasności. Płynnie przechodzące cienie mogą być wytwarzane przez użycie źródeł światła rozproszonego.

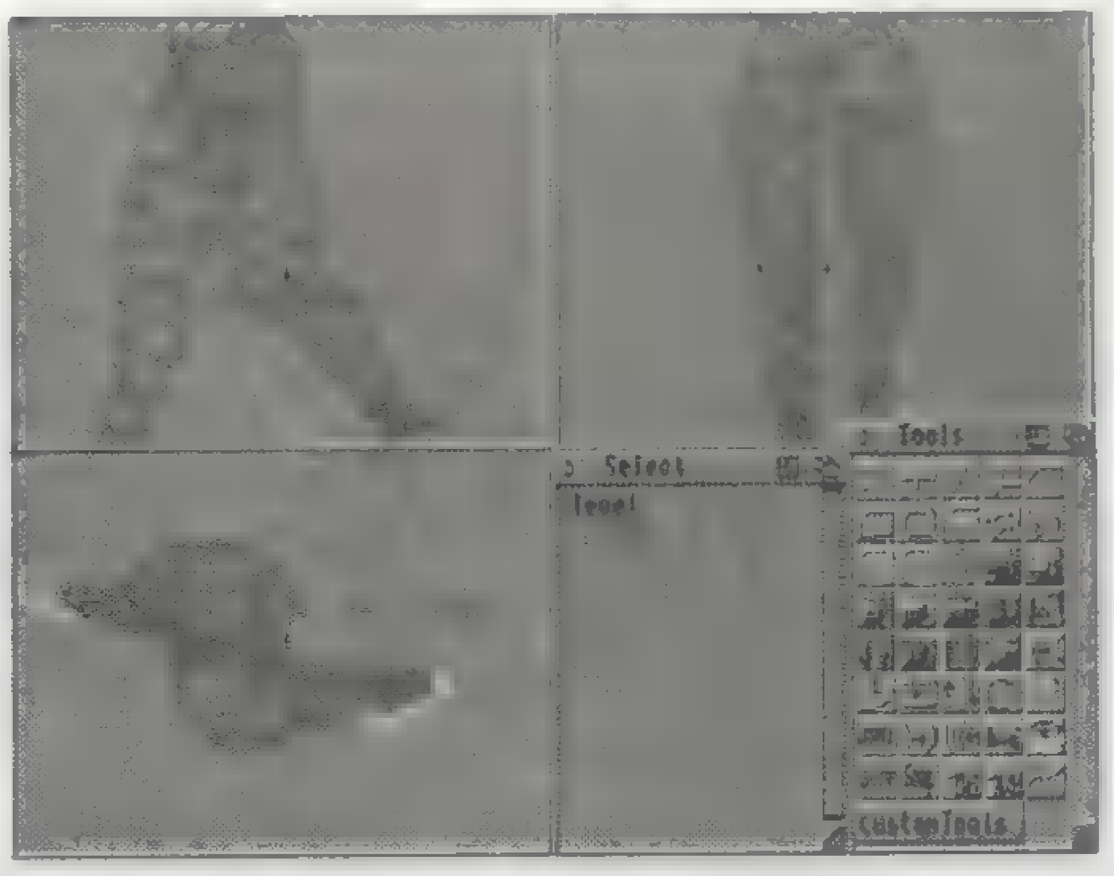
REAL zawiera regulowany ośmio poziomowy anti-aliasing (metoda na wygładzanie "schodków" przy liniach ukośnych, wynikających z niskiej rozdzielczości ekranu). Najdokładniejszy używa oversamplingu 256 * 256 punktów, co jest wystarczające dla każdej rozdzielczości!

Dzięki użyciu wysoko zoptymalizowanego algorytmu, efekt głębii ostrości może być użyty nawet w bardzo złożonych scenach. Nie ma to istotnego wpływu na czas renderingu.

Można także używać efektu ruchu (motion blur) powodującego rozmazanie ruchomego obiektu. Ruch ten odnosi się zarówno do całej sceny, jak i do pojedynczych obiektów.

W celu szybszego wygenerowania obrazu możliwe jest zmniejszenie rozdzielczości. Można podzielić ekran na prostokąty o różnych rozdzielczościach, stopniach anti-aliasingu, itp.

Bardzo ciekawa jest opcja "Outline", która pozwala na wyświetlenie obiektu w postaci kolorowej siatki bez pokazywania niektórych linii. Jest to szczególnie przydatne przy tworzeniu rysunków technicznych lub niezwykle fantastycznych po-



tworków. "Outlines" można także zabarwić i otrzymamy wtedy rysunki perspektywiczne oraz animacje.

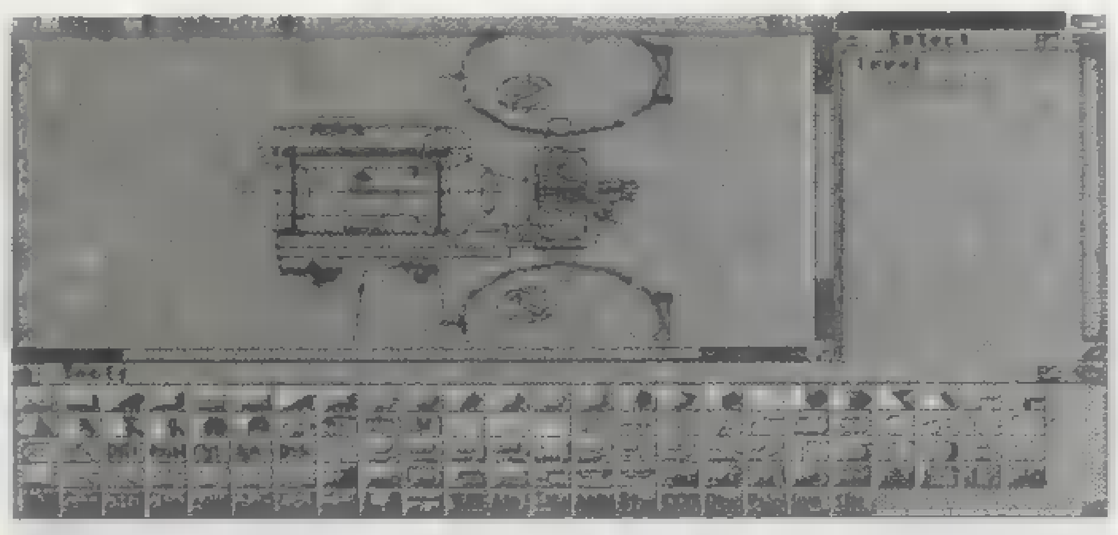
Projektowanie objektów

REAL 3D używa modelowania bryłowego (CSG - Constructive Solid Geometry), co jest najlepszą metodą reprezentowania obiektów trójwymiarowych. Pozwala to na przykład na wykonywanie funkcji logicznych na bryłach i umożliwia prawidłową symulację urządzeń optycznych.

Przy pomocy opcji "Separate I/O" podstawową płaszczyznę w oknie projekcyjnym można obracać luźno w przestrzeni, podobnie jak w spokrewnionym programie "Caligari". Ponadto Real 3D ma do zaproponowania kilka dodatkowych innowacji. Przestrzeń można bowiem przesuwać i poddawać rotacji w sześciu stopniach (na sześciu szczeblach) w stosunku do wszystkich osi przestrzennych, jak również ustawiać według własnego systemu koordynacyjnego.

W REALU obiekty mogą być tworzone w strukturze hierarchicznej. To znaczy, że obiekty są wykonane z podobiektów, a te z dalszych podobiektów itd. Ten rodzaj struktury jest zbliżony do budowy katalogu dysku złożonego z katalogów i podkatalogów. Składowe tych katalogów ułożone są w logiczne grupy dzięki czemu wszelkie modyfikacje są wyjątkowo proste, gdyż można operować całymi grupami obiektów nie zajmując się każdym z osobna. W REALU 3D można posługiwać się obiektami złożonymi tak samo, jak ich częściami składowymi. REAL 3D może wczytywać pliki w formacie DXF. Można bez żadnych problemów wczytywać obiekty z takich programów jak AutoCAD czy z przereklamowanego 3D Studio.

Przy bardzo szczegółowych obiektach "Clip-Boxes" ograniczają szkielet projektu do określonego prostopadłościanu w przestrzeni. Istnieje także możliwość usunięcia przeszkadzających obiektów z pola widzenia w celu większego skoncentrowania się na ważniejszych obiektach. Do



datkowy zysk na szybkości konstruowania jest osiągany przez uczynienie modelu siatkowego niewidocznym. Gotowe obiekty dają się niejako wyłączać. Poza tym można jeszcze określić hierarchię rysowania płaszczyzn i zdefiniować tylko pola szczegółowe. Ta optymalizacja daje największe rezultaty przy projektach z dużą ilością obiektów.

REAL pozwala generować powierzchnie zaokrąglone jak elipsoidy, cylindry, stożki i hiperboloidy. Takie powierzchnie są zdefiniowane matematycznie, co pozwala na rysowanie gładkich figur, nawet bardzo powiększonych. To także przyspiesza działanie programu. Zwiększono także ilość elementów podstawowych. Do dyspozycji stoi ponad 20 figur. Używając operacji logicznych na obiektach, można na przykład podzielić bryłę na dwa niezależne kawałki lub zmieniać rodzaj powierzchni. Dla przykładu używając lśniącego walca można wywiercić lśniącą dziurę w obiekcie matowym. Operacje logiczne są bardzo użyteczne, a nawet niezastąpione.

Różnorodność elementów podstawowych znajduje zastosowanie szczególnie w "Compoud-Tools". Opcja umożliwia budowanie całych scen składających się z wielu objektów. Okrągłe i kanciaste systemy rur, zaokrąglone wielokąty i bryły nie są żadnym problemem, a opcja "Lathe-Tool" umożliwia szybką konstrukcję brył obrotowych. Compound-Tools pracują bardzo efektywnie pod względem wykorzystania pamięci i w związku z tym prace tworzone z ich pomocą są bardzo szybko renderowane. Jednak przy obiektach przezroczystych, jak przedmioty szklane, kanty między poszczególnymi elementami mogą być widoczne. Według informacji Realsoftu kanty te znikają tylko po zmianie stopnia pojemności (objętości - volumenu). Jeżeli jednak obiekty leżą dokładnie obok siebie, nie można tego zmienić. Porządany efekt osiąga się wprawdzie przez przesunięcie obiektów względem siebie, lecz jest to żmudny proces, który wielokrotnie opóźnia tę operację. Kieliszek do wina renderowany przez Lathe-Tool może być o wiele szybciej stworzony niż przy udziale B-Splines.

Poza tym złożone kształty można tworzyć automatycznie z prostych, podstawowych obiektów. Ważną nową cechą jest dodanie krzywych i powierzchni B-sklejanych do konstruowania obiektów freeform (wolnego kształtu). Mogą być one kontrolowane w podobny sposób jak obiekty freeform, lecz ich jakość podczas renderowania będzie znacznie wyższa, niezależnie czy powierzchnia będzie zdefiniowana z niewielkiej ilości punków. Mimo że REAL został zoptymalizowany pod względem obliczania tego typu figur, obliczenia nie są tak szybkie jak powierzchnie cieniowane z użyciem algorytmu Phonga. Jakość jest jednak warta oczekiwania.

Absolutną nowością jest funkcja Mesh i Krzywe-Typ-B-Spline. B-Splines pozwalają na opisanie zakrzywionych swobodnych powierzchni, których forma jest kontrolowana przy pomocy siatki kanciastej (Mesh). Faktycznie taka swobodna powierzchnia nie jest potem interpolowana przez Phong-Interpolation jak to się dzieje przy renderze poligonarnym (Polygon-Render), lecz przez punkty siatki kontrolnej. Kontur B-Spline-Mesh jest przy tym rzeczywiście okrągły i nawet przy największym powiększeniu pozostaje naprawdę bez zarzutu. Czasy renderowania w B-Spline-Mesh są z pozoru wyższe niż czasy ciał poligonarnych (wielokątnych). Wrażenie to jest jednak złudne, ponieważ renderowanie w programach poligonarnych nie daje takiej precyzyjności obiektu.

REAL zawiera wszystkie konieczne funkcje dla modyfikacji prostych kształtów i struktury powierzchni. Jest dostępnych prawie 100 przekształceń nieliniowych i funkcji wyginania. Używając struktury hierarchicznej możliwe jest tworzenie obiektów z różnych materiałów mających odpowiednie własności fizyczne. Wszystkie własności mogą być regulowane bez żadnych ograniczeń. Nawet współczynnik załamania światła jest regulowany dowolnie, co daje możliwość tworzenia przyrządów optycznych jak szkła powiększające. Takie przyrządy zachowują się jak w rzeczywistym świecie - szkło powiększające w REALU naprawdę powiększa!

Użytkownicy, którzy do tej pory pracowali wyłącznie z programami poligonarnymi muszą przyzwyczaić się najpierw do przestrzenno-siatkowego przedstawienia modelu. Kule reprezentowane są tylko przez ich trzy krzywizny równikowe. Wielokąty nie są rozdzielone na wiele trójkątów, a kwadraty są po prostu kwadratami. Pracując przy małej rozdzielczości ekranu lub nad bardzo rozbudowanymi (kompleksowymi) projektami można się szybko przekonać, że ta metoda przedstawiania modeli ma wiele zalet. Okna edytorskie są bardziej uporządkowane i przejrzyste niż w innych programach, a czas budowy modelu siatkowego odpowiednio krótszy.

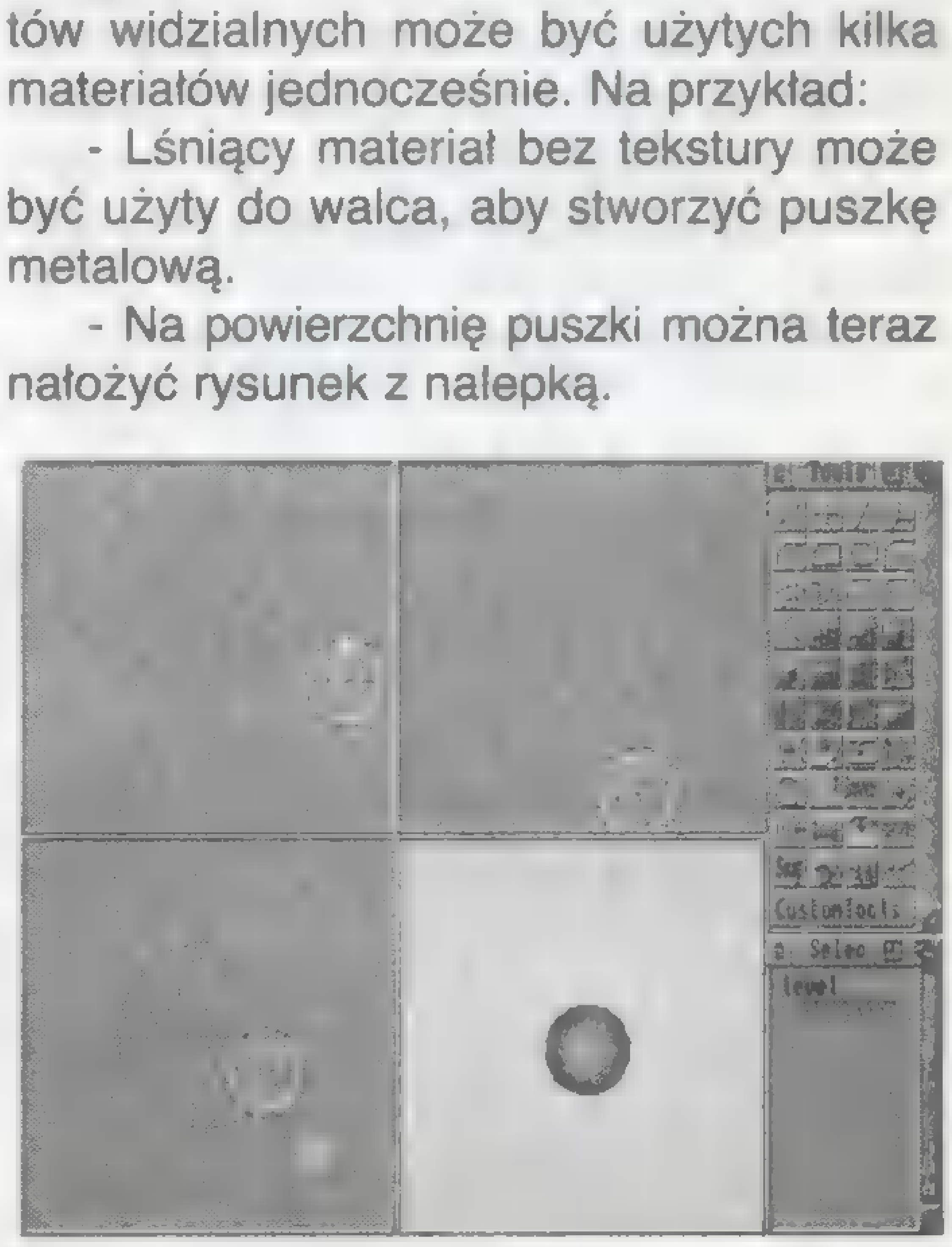
Język RPL

Nowością jest język programu RPL (Real 3D Programming Language). Pozwala on na sterowanie wszystkimi parametrami obiektu, animacji i materiałów. Jest on także wykorzystywany do opisu makrodefinicji i do eksportowania scen jako ASCII. To ostatnie umożliwia także wpisanie konwertera do obcych programów. Przy udziale RPL można tworzyć metody animacji i programowalne tekstury. Bardzo korzystny jest fakt, że dzięki RPL istnieje możliwość rozszerzenia edytora i dopasowania go do własnych celów. Nowa wersja REALA posiada też własny port ARexxa. Dzięki niemu możliwe jest przekazywanie przez inne programy poleceń w RPL dla Reala 3D. Dostępne są dwa formaty: binarny i ASCII. Format binarny daje szybki i efektywny sposób na przechowywanie opisu sceny, natomiast format ASCII bazujący na języku RPL, jest zależny od maszyny.

Edytor materiałów

Edytor materiałów służy do definiowania wszystkich parametrów materiału. Można tym samym ustawić wielkość i kolor połysku, stopień odblaskowości, przezroczystość, załamanie światła, zamglenie i zależność tych zjawisk od objętości obiektu (przy efektach gazowych i zamgleniach), cząsteczkową chropowatość powierzchni i ciężkość materiału. Te ostatnie są konieczne, kiedy model składa się z kilku zmieszanych lub stopionych tworzyw. Materiały są związane z obiektami w hierarchii w ten sam sposób jak metody. Mają także wpływ na obiekty o tym samym poziomie. Oznacza to, że do obiektów widzialnych może być użytych kilka materiałów jednocześnie. Na przykład:

- Lśniący materiał bez tekstury może być użyty do walca, aby stworzyć puszkę metalową.
- Na powierzchnię puszki można teraz nałożyć rysunek z nalepką.

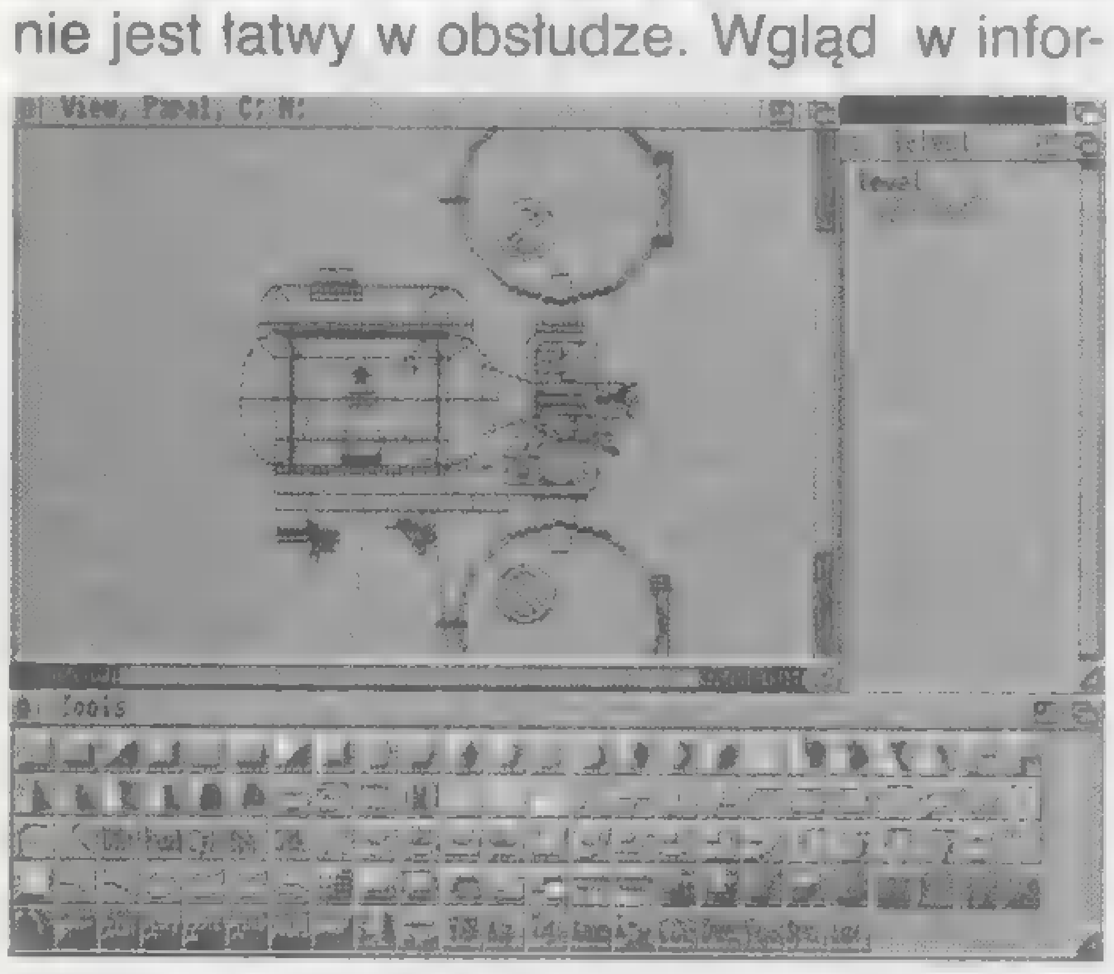


- Trzecim materiałem można utworzyć wystające krawędzie puszki poprzez nałożenie mapy pofałdowań z rysunkiem matematycznie zdefiniowanego koła.

Tekstury

Tekstury mogą mieć reprezentację geometryczną na ekranie edytora, która może być modyfikowana jak każdy inny kształt geometryczny. Mogą one zmieniać pozycję i wymiary tak, aby idealnie pasowały do projektu. Oczywiście tekstura jak każdy obiekt może być animowana! Praktycznie wszystkie parametry obiektu można określić przez teksturę. Komu to nie wystarcza, ten może sięgnąć po tekstury proceduralne, matematyczne formuły tekstur i po tekstury własnoręcznie zaprogramowane. RPL umożliwia tu wszystkie możliwe do wymyślenia warianty. Zdefiniowane formuły tekstur umożliwiają utworzenie między innymi falistych "Bumpmaps". "Shadow mapping" oblicza malowanie cieni na powierzchni. Przynosi to ogromną oszczędność czasu kalkulacji. "Spline-mapping" nakłada tekstury jak drugą skórę na B-Spline-Meshes. Wszystkie te udogodnienia najlepiej doceni ten, kto choć raz próbował odtworzyć ruch trzepoczącej flagi. Opcje "Tiling" i "Filiping" umożliwiają nałożenie tekstur na większe lub niekończące się powierzchnie. "Gradient" interpoluje kolor między punktami tekstury. "Clip Mapping" i "Transparent Color" umożliwiają wykrojenie koloru z tekstury, aby otrzymać prawdziwe (rzeczywiste) otwory w obiektach. Obiekty typu "Matte" pozwalają na zastosowanie danych o głębokości obrazu w tle.

Kształtownie tekstur (handling) pozwala na osiągnięcie niemal niewyobrażalnych kreacji I uzyskiwanie rzeczy dotychczas niewykonalnych. Ponieważ tekstury pierwotne łatwo można poddać animacji, nie stwarzają już kłopotu animacje materiałowe kosztujące zazwyczaj wiele wysiłku. Istnieje tylko jeden wymóg do spełnienia - opanowanie wszystkich parametrów edytora. Wymaga to pewnej rutyny i czasu, ponieważ edytor materiałów nie jest łatwy w obsłudze. Wgląd w informator znacznie pomaga przy ustawianiu parametrów materiałowych i jest w każdym razie godny polecenia.



System animacji

REAL zawiera zupełnie nowy system animacji. Animacje nie bazują już na klatkach i kolejno eksponowanych obiektach. Teraz istnieje globalna wartość kontrolująca jak mają być animowane obiekty i specjalne okno do sterowania czasem. Czas jest ciągły i w dowolnym jego momencie można obejrzeć stan i pozycje obiektów. Chwile czasu albo klatki stają się animacją dopiero podczas renderowania.

Efekty animacyjne (zwane metodami) t.j. PATH, SWEEP, COLLISION itd. są teraz dołączone do obiektów w hierarchii. Metody te mogą oddziaływać na wszystkie obiekty o tym samym poziomie w hierarchii, a niektóre mogą oddziaływać na inne metody. Znaczy to, że możliwe jest zbudowanie animacji składającej się ze złożonych obiektów etapami:

- zrobienie nogi poruszającej się w biodrze.
- zrobienie stopy poruszającej się na nodze.
- zrobienie całej nogi biegnącej po ścieżce.

Cały ten obiekt może być umieszczony gdziekolwiek w przestrzeni i noga będzie poruszała się sama podczas animacji. Wszystkie metody mogą być kopiowane i zmieniane przez funkcje modyfikacji jak każdy inny obiekt. Jeśli ich efekty animacyjne będą kontrolowane przez inne obiekty o jakiejś geometrii, to ich efekty będą modyfikowane razem z nimi. Tak więc biegnąca noga może być powielona, aby utworzyć parę, a z paroma modyfikacjami wymiaru i pozycji, nogi mogą stać się rękami biegnącego człowieka. Te animowane obiekty mogą być zapisywane przez użytkownika. Komu nie wystarcza liczba i rodzaj wyżej wymienionych metod animacji, temu RPL pozostawia wolną rękę w programowaniu własnych, nowych metod.

Absolutnie nowy jest system "Particle-Animation". Dzięki niemu można, przy małym wysiłku, automatycznie dokonywać animacji równocześnie wieloma obiektami, których ruch uwarunkowany jest prawidłami fizycznymi jak prędkość, spin, masa i tarcie. Spadające piłki, które zderzają się ze sobą i ponownie odbijają się od ziemi są generowane błyskawicznie. Odpada przy tym proces żmudnego kreowania ścieżek i przebiegów. Tego rodzaju tricki są dostępne jak dotąd tylko dla drogich wysoko wyspecjalizowanych



stacji graficznych. Aby umieścić dodatkowe informacje jak ciężar, kierunek obrotu, szybkość i wiele innych potrzebnych szczególnie przy animacjach cząstkowych istnieje w REALU "TAG-System", przy pomocy którego dla określonego przedmiotu zwiększa się charakterystyczne dla niego dane, rozszerza jego strukturę.

W niektórych opcjach obsługa programu wymaga ulepszenia. Często zachodzi bowiem konieczność własnoręcznego (ręcznego) podawania parametrów. Czasowa kontrola przebiegu animacji powinna zostać uproszczona przez wygodny requester. "Color" requester wymaga pilnego poprawienia. Wadą jest także brak możliwości exportu na inne formaty obiektów oraz wadliwy moduł importu plików .DXF. W programie istnieją punktowe, liniowe i powierzchniowe źródła światła służące do oświetlania scen. Dwa ostatnie pozwalają na wprowadzenie cieni, co wielokrotnie wydłuża czas renderowania. Brakuje jednak zwyczajnego oświetlenia miejscowego, co jest dużą wadą.

Biorąc pod uwagę zakres możliwości program jest bezkonkurencyjny. Konkurentów dla Reala należy szukać tylko na specjalizowanych stacjach graficznych, gdyż pozostałe programy amigowskie jak i proto PeCeTowskie nie dorastają mu do pięt. Z punktu widzenia możliwości programu cena 1000 DM jest niewiarygodnie korzystna. Otrzymujemy w pełni profesjonalne narzędzie za bardzo niską cenę.

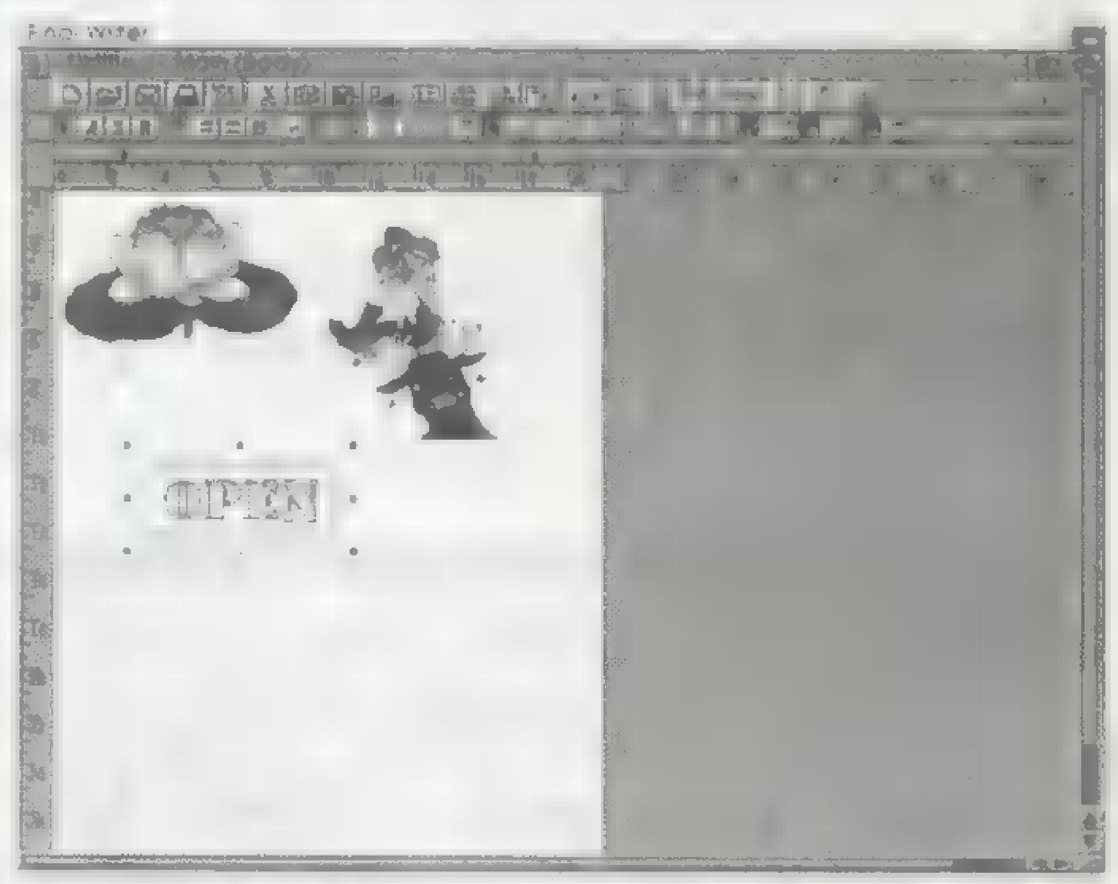
Z ostatniej chwili

Zostały już ukończone pracę nad wersją Pecetowską REALA v.2.0. Wygląda on tak samo jak na Amidze i wszystkie funkcje renderowania są wykonywane z taką samą dokładnością. Wersja ta jest podzielona na odrębne bloki jak ma to miejsce w programie Soft Image 4D na Silicon Graphics. I tak chcąc np. wgrać brush w formacie TARGA musimy dokupić oddzielny loader (dla ciekawości loader TARGA do programu Soft Image kosztuje około 1000 USD).

Final writer

Krzysztof Nowicki

SoftWood to ciekawa firma, która namiętnie w nazwach niektórych swoich produktów używa słówka Final, lecz wcale nie zamierza spocząć na laurach i rozwija ten swój Final nadal. Przykład: Final Copy II i jego następca Final Writer. SoftWood to firma nietypowa. Jej nietypowość polega na tym, że w odróżnieniu od większości firm zachodnich, nie reklamuje swojego produktu zanim nie jest ukończony.



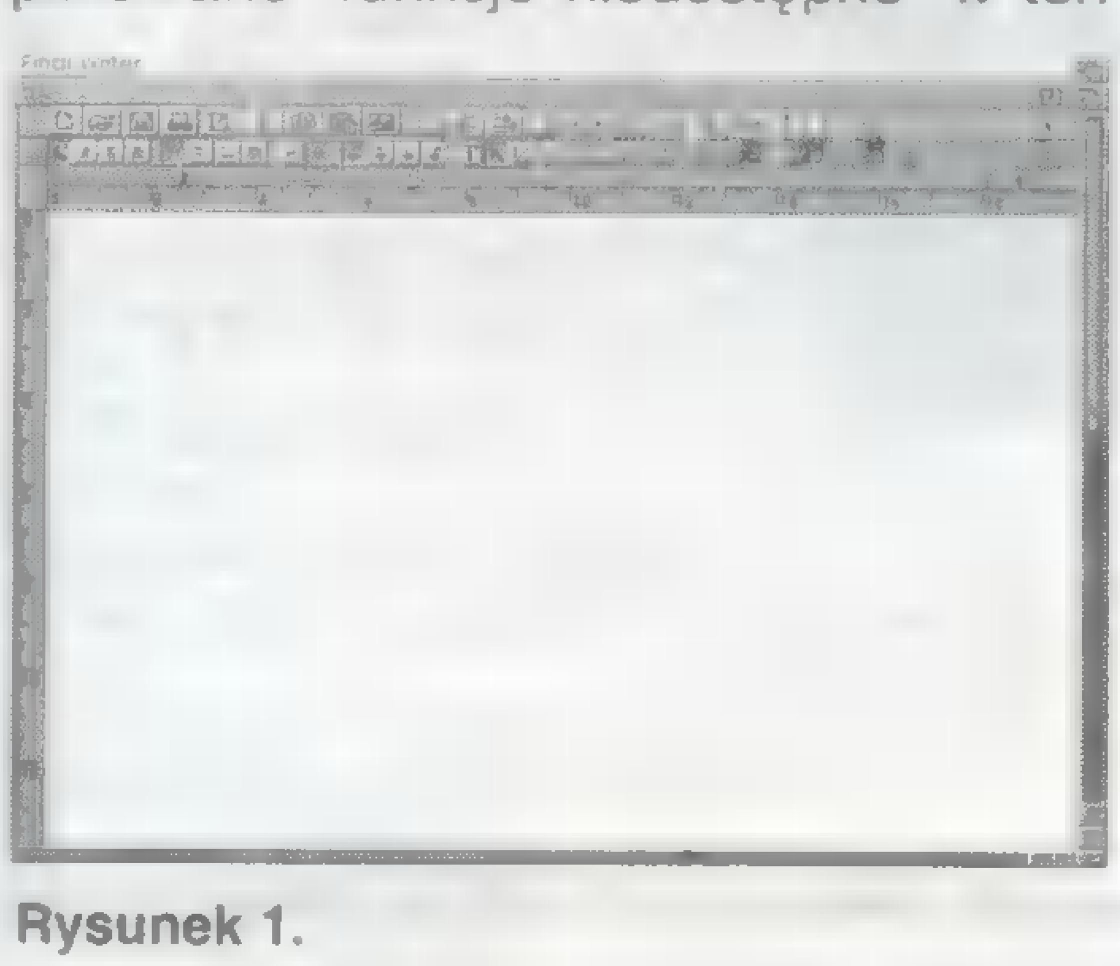
Rysunek 4.

Nowy produkt tej firmy to skrzyżowanie procesora tekstu i programu DTP. Nosi on wspomnianą już nazwę Final Writer I jest rozwojową wersją znanego edytora Final Copy. W porównaniu do swojego poprzednika jest to program bardzo rozbudowany i swoimi możliwościami dorównuje znanym już (i drogim) edytorom opracowanym dla IBM PC i Macintosha.

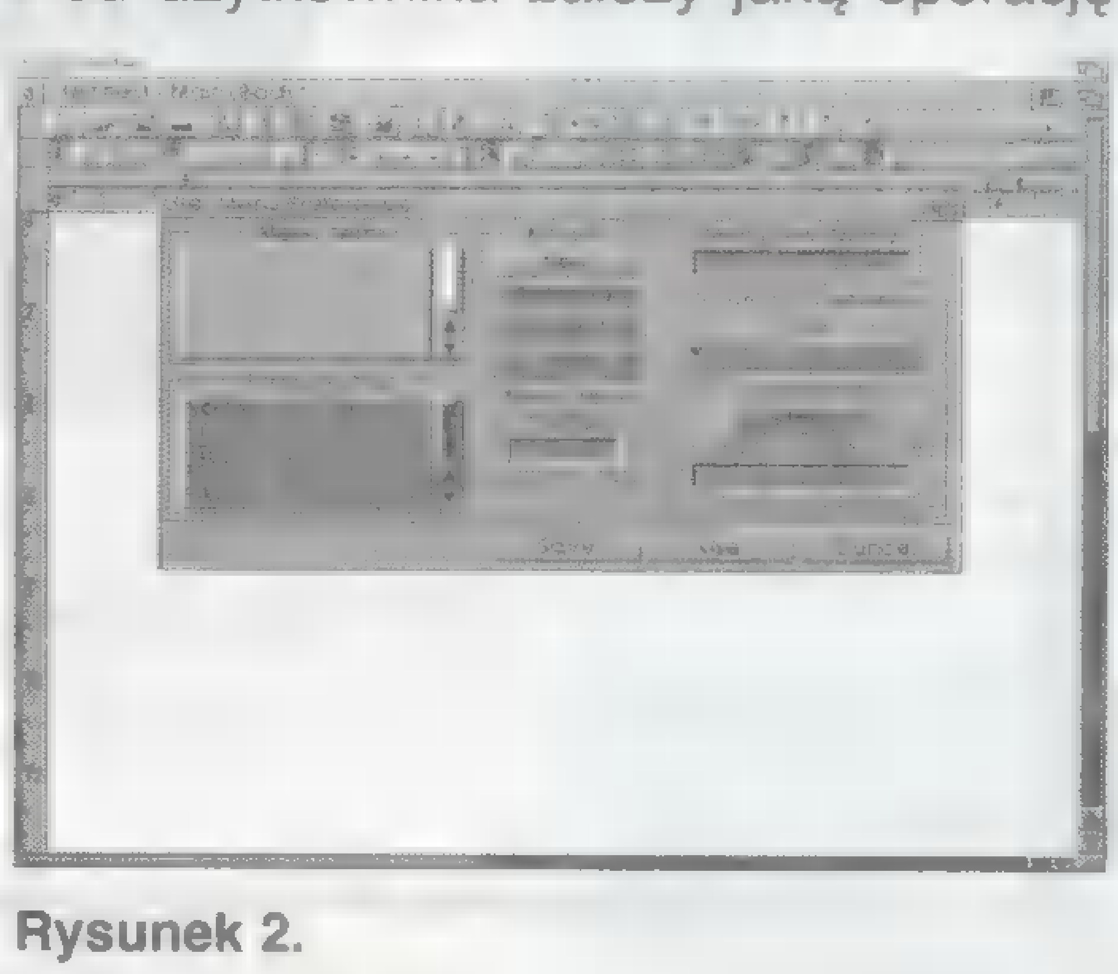
Przy projektowaniu systemu komunikacji z użytkownikiem połączono doświadczenia programistów zajmujących się tworzeniem oprogramowania na dwa wspomniane wyżej komputery (w przypadku IBM dotyczy to oczywiście wyłącznie środowiska Windows). Dlatego na ekranie oprócz standardowych, znanych z Final Copy gadżetów i menu, pojawiają się niespotykane dotychczas ikonki realizujące nieraz bardzo skomplikowane funkcje niedostępne w ten sposób w żadnym ze znanych dotychczas edytorów na Amigę. Efektem tego jest upodobnienie wyglądu i sposobu obsługi Final Writera do programów typu Ami Pro lub Word for Windows. (Patrz Rysunek 1.)

Oprócz samych ikonek użytkownik otrzymał od programistów w prezencie możliwość dowolnego manipulowania ich wyglądem, pozycją na ekranie oraz związanymi z nimi funkcjami. (Patrz Rysunek 2.)
(Patrz Rysunek 3.)

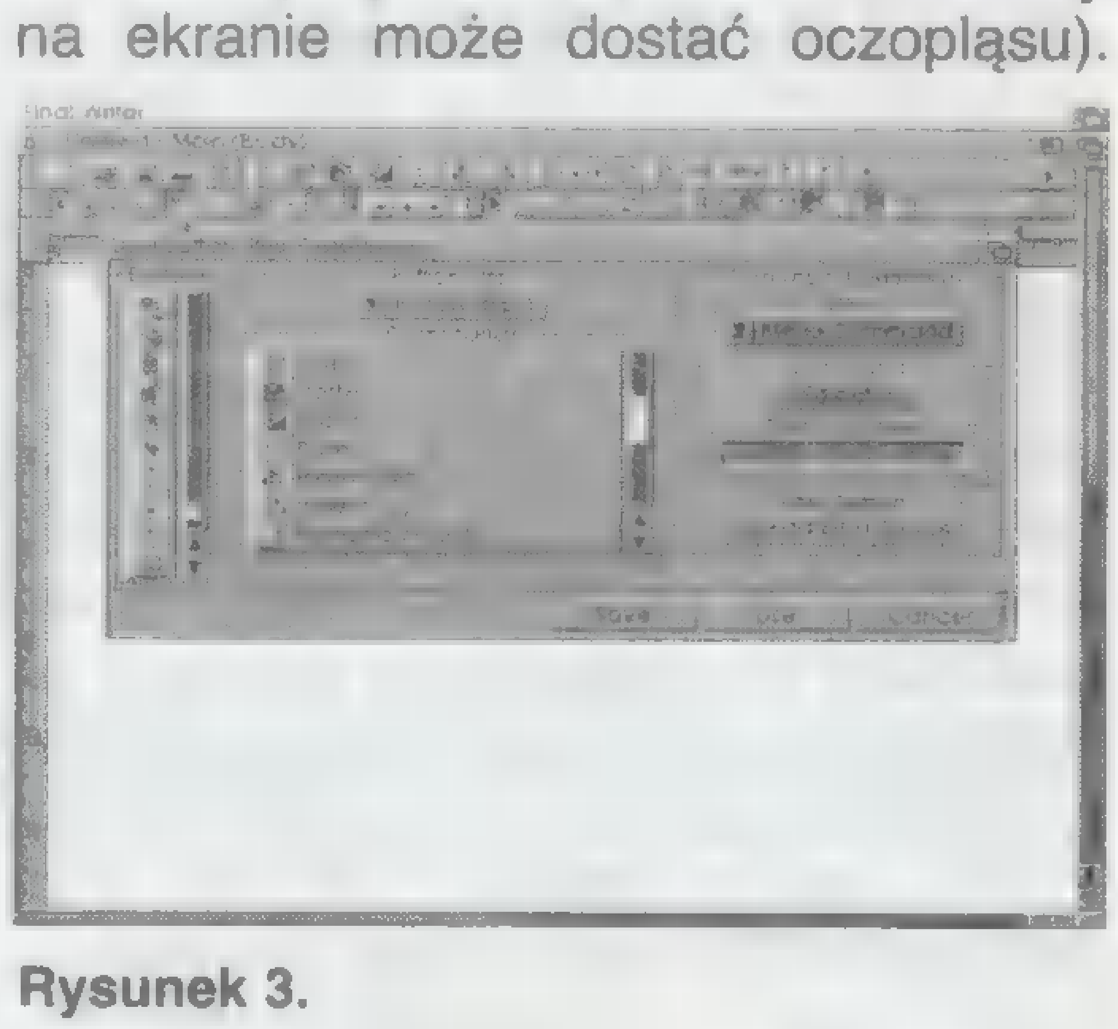
Do wyboru mamy aż 180 takich ikonek, które można rozmieścić w nagłówku okna programu w 8 bankach. Niektóre spośród tych ikonek mają z góry przypisane funkcje np. nagrywanie lub odczytywanie dokumentu, drukowanie, zmiana strony, narzędzia graficzne, zmiana fontu, itd. Inne zaś są "czyste" i od użytkownika zależy jaką operację będą wykonywały. Funkcje realizowane przez te pseudo przyciski nie ograniczają się zresztą tylko do obsługi programu Final Writer. Dysponują one bowiem oknem na świat w postaci portu Arexxa. Możliwe jest więc przypisanie ikonce programu w języku Arexx lub też komendy AmigaDOSu. Jakie ma to znaczenie nie trzeba chyba tłumaczyć. Odpowiednio "zaprogramowaną" ikonkę można także wykorzystać do bardziej przyziemnych celów np. do umieszczania w tekście swojego imienia i nazwiska wraz z adresem. Te olbrzymie możliwości wymagają pewnego usystematyzowania wynikającego nie tylko z ich ogromu, lecz również wielkości ekranu. Wydaje się jasne, że umieszczenie 180 ikonek na jednym ekranie nie jest możliwe (normalny człowiek przy takim ładunku informacji na ekranie może dostać oczopląsu).



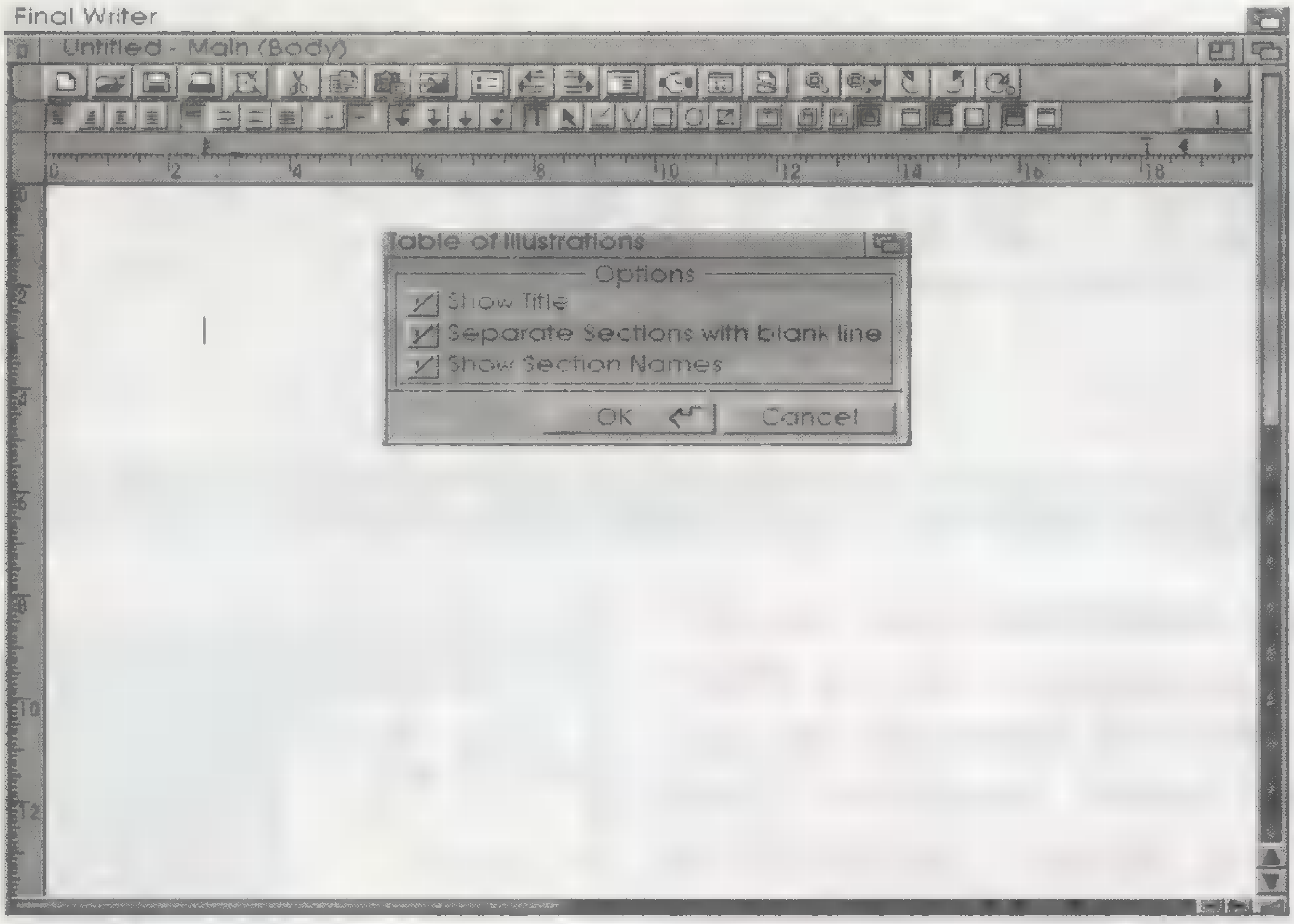
Rysunek 1.



Rysunek 2.



Rysunek 3.



Rysunek 5.

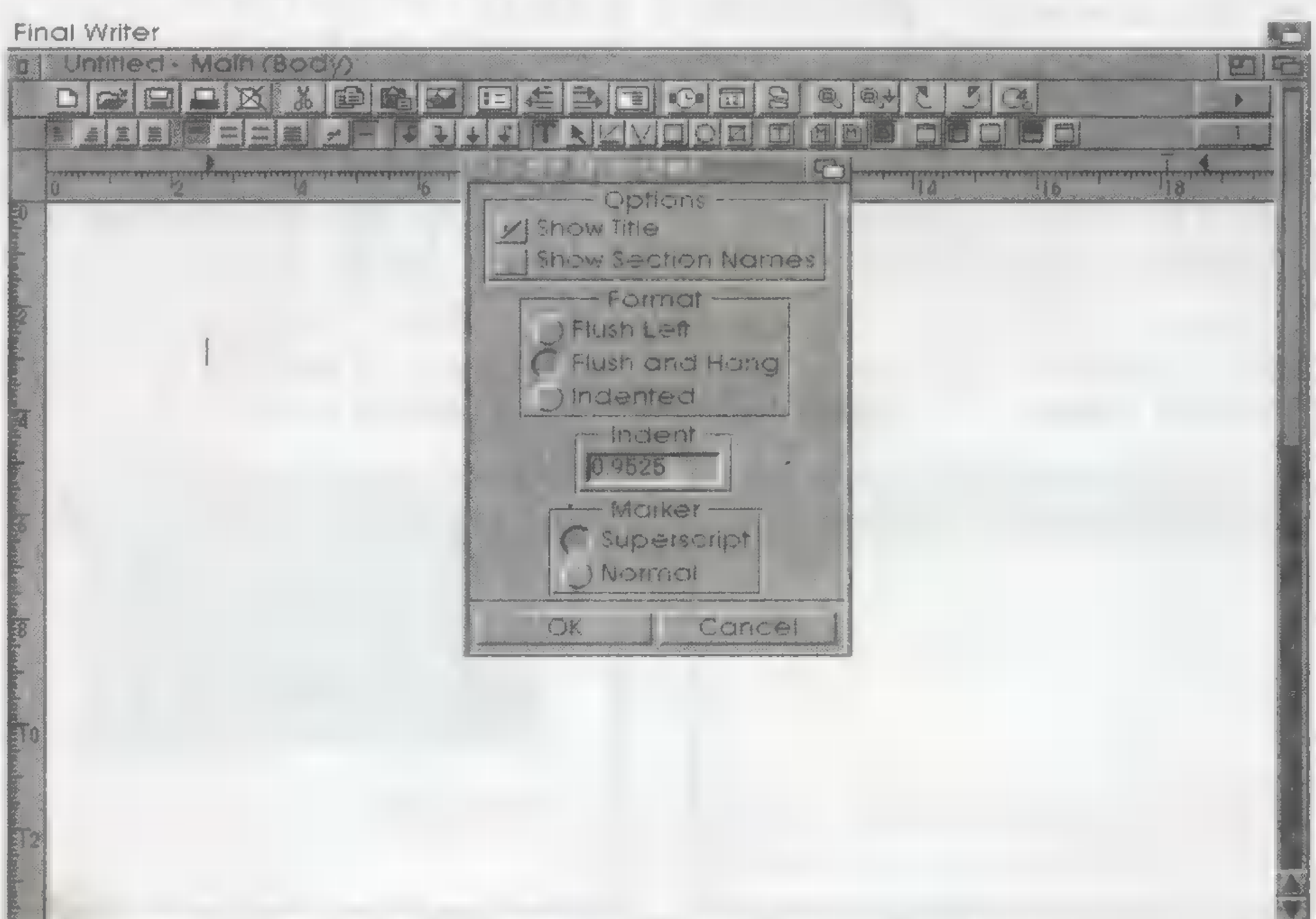
W związku z tym ikonki mogą być podzielone na 8 grup, z których tylko jedna jest widoczna w danym momencie. Pozwala to np. na zgrupowanie ikonek o działaniu obejmującym cały dokument (odczyt, zapis, tworzenie nowego dokumentu, drukowanie, preferencje) w jednej grupie, zawierających funkcje graficzne w drugiej, operacje na tekście w trzeciej, itd.

Pojawienie się nowego interfejsu graficznego nie obejmuje oczywiście wszystkich zmian jakim poddano Final Copy, aby przekształcić go w Final Writera. Zmiany te są bowiem o wiele bardziej znaczące. Pierwszą, którą chciałbym wymienić jest pojawienie się nowego filtra graficznego pozwalającego umieszczać w dokumentach nie tylko grafiki bitowe w standardzie IFF, lecz również grafikę wektorową nagraną jako plik EPS. (Patrz Rysunek 4.)

Tworzenie obrazu w tym formacie umożliwia program Art Expression firmy Soft Logic oraz praktycznie wszystkie programy dla IBM-a i Macintosha. Skutkiem ubocznym wprowadzenia tego filtru stała się możliwość wykorzystywania dokumentów innych programów nie związanych ze światem Amigi (nic nie stoi na przeszkodzie, aby przekonwertować sobie bibliotekę graficzną Corel Draw). Kontynuując wątek standardu EPS trzeba dodać, że Final Writer umożliwia nam dokonanie wysokiej jakości wydruku tak zapisanej grafiki nie tylko na drogich drukarkach postscriptowych, ale także na zwykłych - mozaikowych (z czym wiele programów ma poważne problemy, chociaż w ich prospektach reklamowych wymienia się taką funkcję). Nowym elementem związanym z obróbką grafiki jest możliwość obracania okienka zawierającego obraz o dowolny kąt, co było dotychczas domeną programów DTP. Ze świata DTP pochodzi też funkcja pozwalająca na grupowanie kilku obiektów graficznych w jedną całość. Oczywiście zachowano stare funkcje znane z Final Copy tj. skalowanie, przesuwanie oraz wycinanie grafiki.

Nie zapomniano oczywiście o głównym zadaniu programu tj. edycji tekstu. Final Writer zawiera w sobie wszystkie podstawowe funkcje związane z edycją tj. możliwość wyboru czcionki, jej wielkości, atrybutów, odległości między liniami, itp. Poprawiono nieco i tak już doskonały słownik ortograficzny oraz słownik wyrazów bliskoznacznych (niestety dostępny tylko w wersji angielskiej). Równie ciekawym przemianom poddano funkcje automatycznego tworzenia indeksów do tekstu zgodnie z nawet najbardziej wymyślnymi życzeniami użytkownika. Całkowitą nowość natomiast stanowią nowe funkcje pozwalające wygenerować w sposób automatyczny spis treści, notę końcową lub bibliografie. (Patrz Rysunek 5.)

Final Writer umożliwia także automatyczne wygenerowanie spisu zawartych w aktualnie obrabianym dokumencie ilustracji. Oczywiście na tym nie koniec. Dostępne jest automatyczne dzielenie wyrazów (niestety, ale nie obejmuje niektórych wyjątków związanych z polską pisownią). Ciekawym elementem jest zastosowanie znanych w programach DTP stron wzorcowych, które umożliwiają tworzenie bardzo długich dokumentów. Idea strony wzorcowej opiera się o fakt, że kształt graficzny poszczególnych stron w długich dokumentach nie różni się od siebie w sposób znaczący. Strona wzorcowa zawiera wszystkie elementy wspólne. W ten sposób nie jesteśmy zmuszeni do umieszczania na każdej kartce dokumentu np. linii dzielącej kolumny lub numeru strony. Z pomocą Final Writtera operację tę możemy przerzucić na barki komputera.



Final Writer jest dostarczany w pakiecie zawierającym oprócz samego programu 120 fontów wektorowych w formacie Postscript, 100 przykładowych obrazów wektorowych w formacie EPS oraz 25 skryptów Arexxa.

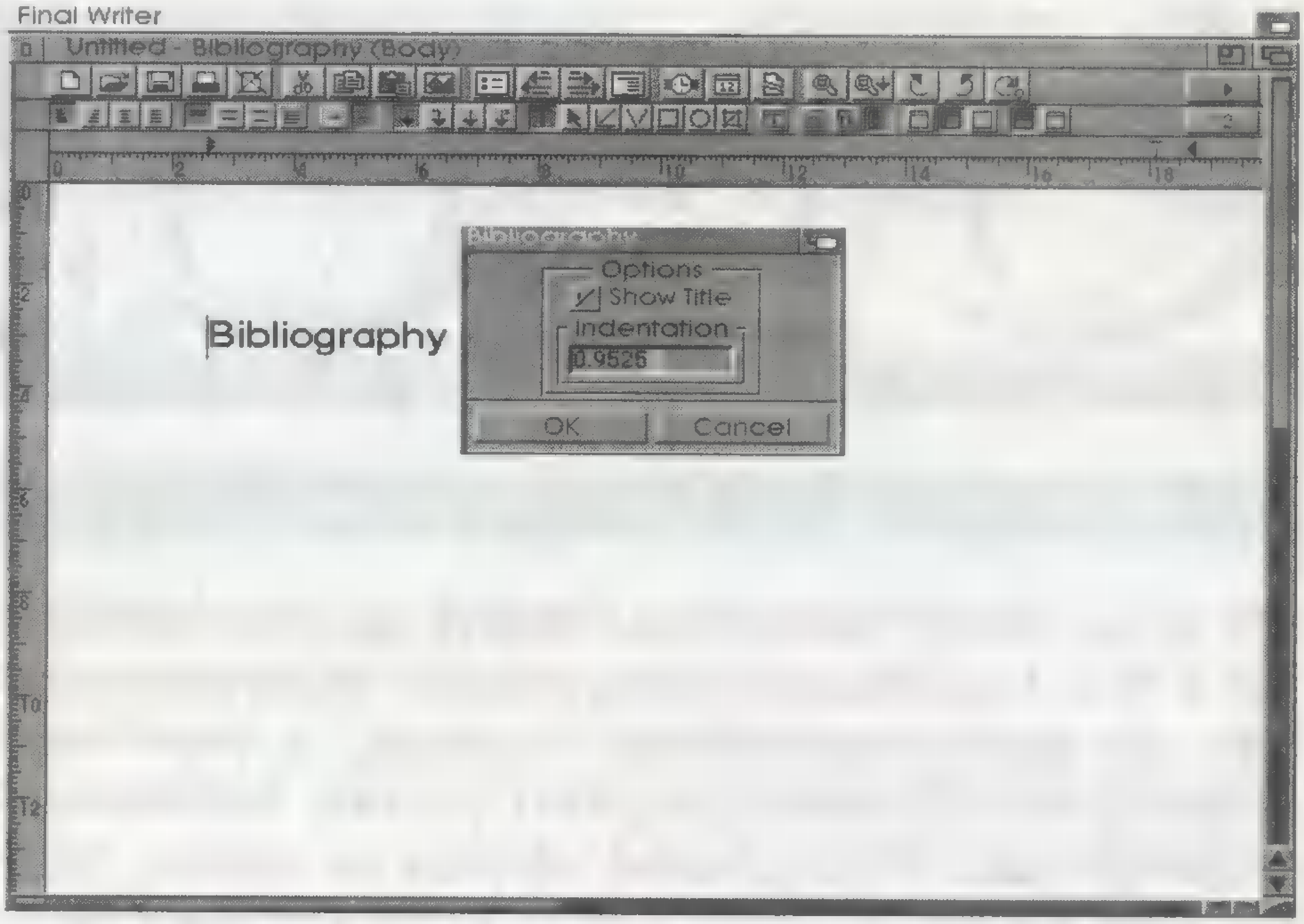
Dla polskiego użytkownika ważny jest z pewnością problem czcionek. Firma SoftWood nie produkuje jak na razie polskiej wersji językowej swojego programu, więc nie trudno się domyślić, że Final Writer nie dysponuje polskimi znakami narodowymi. Z pewnością zainteresuje potencjalnego nabywcę fakt, że edytor ten rozpoznaje takie formaty czcionek jak Compugraphic (standardowy format czcionek wektorowych w systemie Amigi) oraz NimbusQ (stosunkowo mało znany). Pozwala to na zastosowanie łatwo dostępnych polskich fontów wektorowych w tych formatach (szczególnie tym pierwszym).

Warto jeszcze na zakończenie tego hymnu pochwalnego przypomnieć kilka faktów znanych użytkownikom Final Copy. Final Writer podobnie jak jego poprzednik Final Copy zachował w 100% niesamowitą prędkość działania. Porównanie szybkości dysponującego znacznie mniejszą paletą funkcji Pro Write z nowym produktem SoftWood, daje w wyniku kilkukrotną różnicę w czasie wykonywania większości operacji na korzyść FW. Warto zauważyć, że Final Writer jest jednym z niewielu programów, które radzą sobie ze wszystkimi stosowanymi w Amigach systemami operacyjnymi oraz układami graficznymi. Jest to także drugi edytor (po Final Copy), który nie tylko działa z układami AGA, ale w pełni je wykorzystuje. Jeżeli będziemy pracowali w trybie 256 kolorowym to wczytana do programu grafika będzie pokazywana w 256 kolorach, a nie w 4 czy 16. O wysokiej jakości Final Writera mogą się przekonać właściciele kart graficznych. Program ten ze względu na możliwość pracy jako okno na ekranie Workbencha, pozwala uruchomić się pod wszystkimi emulatorami Workbencha dostarczanymi z większością kart graficznych.

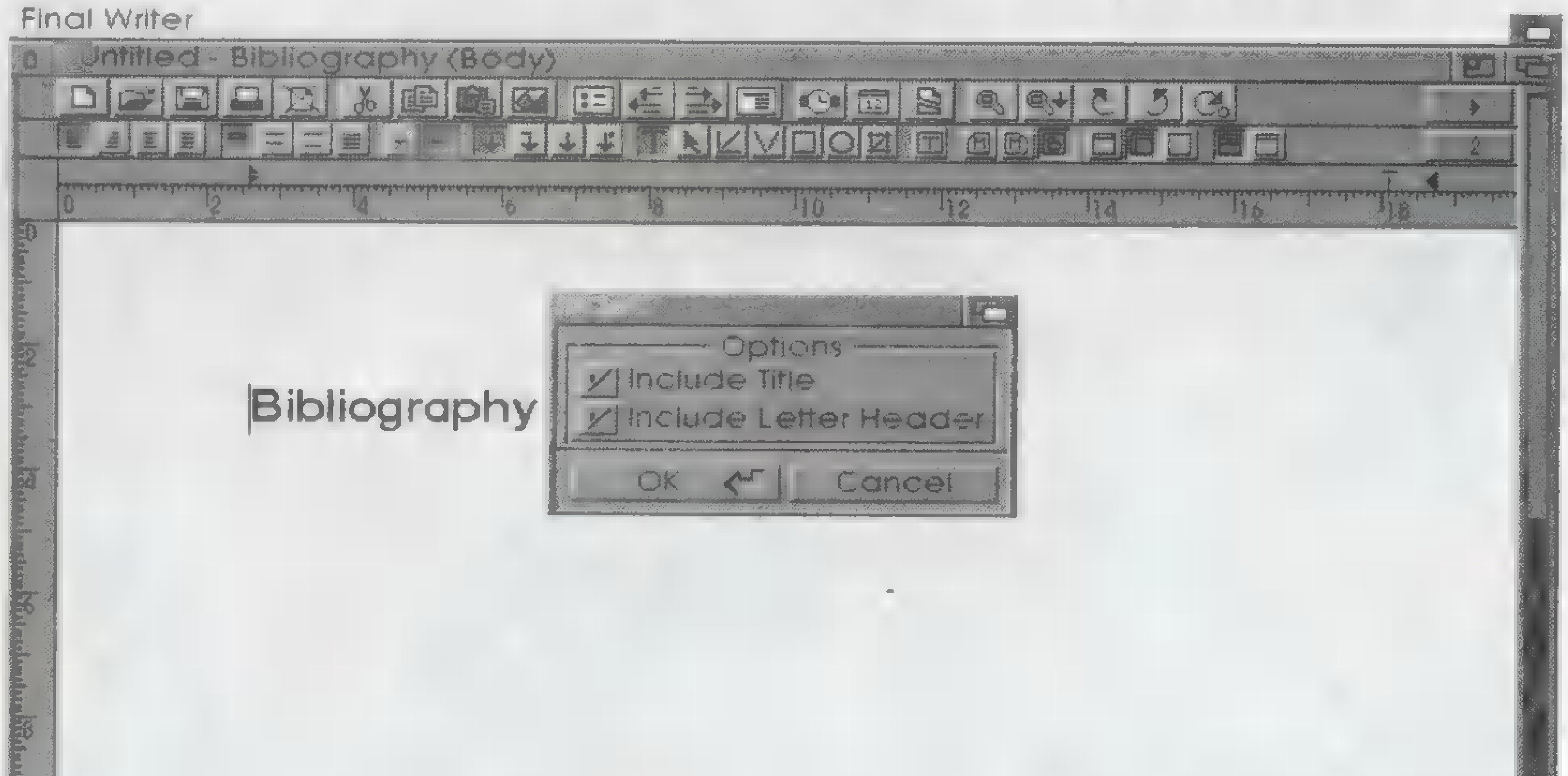
Oczywiście Final Writer nie jest pozbawiony wad. Podstawową jest oczywiście dla polskiego użytkownika brak odpowiedniej wersji językowej. Ponadto w samym programie zauważyliśmy bardzo poważne zaniedbanie (występujące również w Final Copy) jest nim brak funkcji UNDO. Wydaje mi się, że program o tak "profesjonalnych" ambicjach powinien taką operacją dysponować. Innym mankamentem, który dotyczy nie tylko tego edytora, ale wszystkich, które opracowano dotychczas dla Amigi, jest brak możliwości szybkiej edycji w programie tabelek oraz równań matematycznych. Bez tych dwóch funkcji program nie traci co prawda na wartości, ale w oczach użytkownika piszącego teksty wyłącznie "humanistyczne" traci wiele. Trudniej będzie wzbudzić zainteresowanie wśród ludzi, którzy swoje teksty opracowują z wykorzystaniem matematyki lub też opierają np. o dane statystyczne. Brak możliwości szybkiego wykonania tabelki lub napisania wzoru bez uciekania się do programów typu Art Expression lub Pro Page dyskwalifikuje taki edytor w sposób automatyczny.

Chcąc podsumować tę krótką recenzję trzeba stwierdzić, że Final Writer znacznie podnosi poprzeczkę jakości dla edytorów tekstu, a także przesuwa ją nieco w kierunku programów DTP. Jest to produkt godny polecenia wszystkim posiadaczom Amigi. Biorąc pod uwagę jego szczególną szybkość wart jest uwagi, tych spośród użytkowników, którzy dysponują powolniejszymi modelami naszej "Przyjaciółki". Trzeba jednak zaznaczyć, że pazury program pokazuje dopiero wtedy, gdy dysponujemy 2 MB RAMu oraz dyskiem twardym. ❑

Final Writer



Final Writer



OPUS v4.x cz.1

Tomasz Kokoszczyński

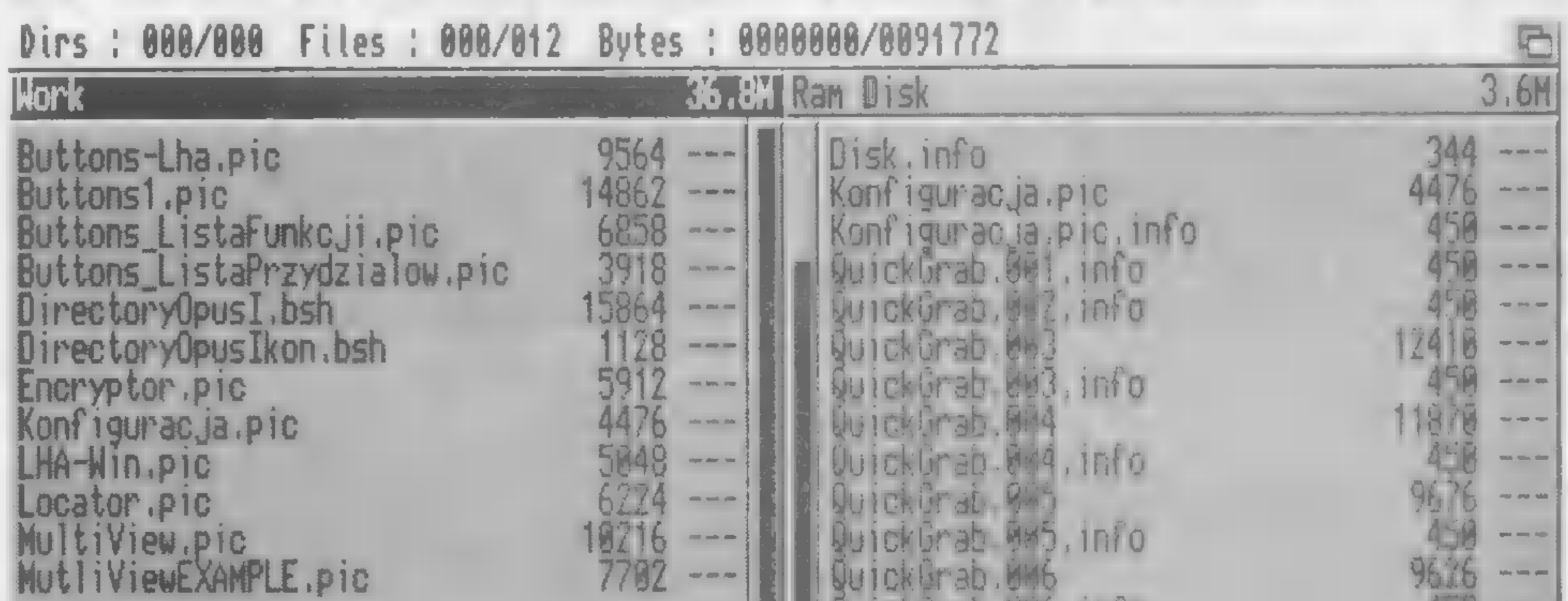
Chyba każdy amigowiec zdążył się już zetknąć z tym legendarnym programem. Możliwościami bije na głowę normalnego Nortona, a jego najnowszej wersji ustępuje (sic!) jedynie wielkością (zajmuje coś z 6 razy mniej miejsca na dysku). Jest to jeden z lepiej przemyślanych programów na Amigę i na dodatek chyba jeden z najczęściej używanych... Dziś zajmiemy się Opusem praktyczniej.

Swego czasu byłem piętnowany w redakcji za promowanie Opusa. Twierdziłem, że taki program nie może się przeżyć i że nawet najbardziej genialny system typu "Workbench" nie zastąpi go, ponieważ Opus jest jego uzupełnieniem. Niektórzy z moich kolegów zaczęli nawet ostentacyjnie ignorować Opusa używając do kopiowania plików komend systemowych. Nie trwało to jednak długo. W obecnej chwili praktycznie zawsze, gdy odwiedzam redakcję i dochodzi do kopiowania plików do akcji wchodzi Opus. Program jest bardzo łatwy i wygodny w obsłudze. Posiadacze Amigi 500 wyposażeni w 1MB pamięci mogą być zawiedzeni niewielką ilością pamięci, która pozostaje po odpaleniu Opusa. Niestety - program zajmuje ponad 500 KB pamięci i może się dać nawet we znaki przy pracy z 4MB przy korzystaniu z innych pamięciożernych programów. To wszystko prawda. Zgoda, że system buforowania wszystkich wczytanych katalogów może jeszcze bardziej powiększyć zajmowaną pamięć. Jednak uważam, że warto używać Opusa, gdyż jest niewiele programów, z którymi pracuje się tak przyjemnie.

W niniejszym opisie zajmiemy się najnowszą wersją Opusa, jaka pojawiła się ostatnio na rynku, czyli v 4.11. W wersji tej zmieniło się bardzo wiele i zarazem tak niewiele w stosunku do opisywanej przez nas wersji 3.41 (AMIGOWIEC 10-11/92, str. 20-23). Drobne, kosmetyczne wręcz zmiany panelu głównego jeszcze bardziej poprawiły wygodę użytkowania. Większe zmiany nastąpiły w modułach konfiguracyjnych. Skonfigurowanie programu jest teraz o wiele łatwiejsze i daje o wiele więcej możliwości niż w wersji poprzedniej. Dodatkowo wprowadzono możliwość "zewnętrznego" kopiowania plików, dyskietek, czy formatowania dyskietek - bez zajmowania samego Opusa.

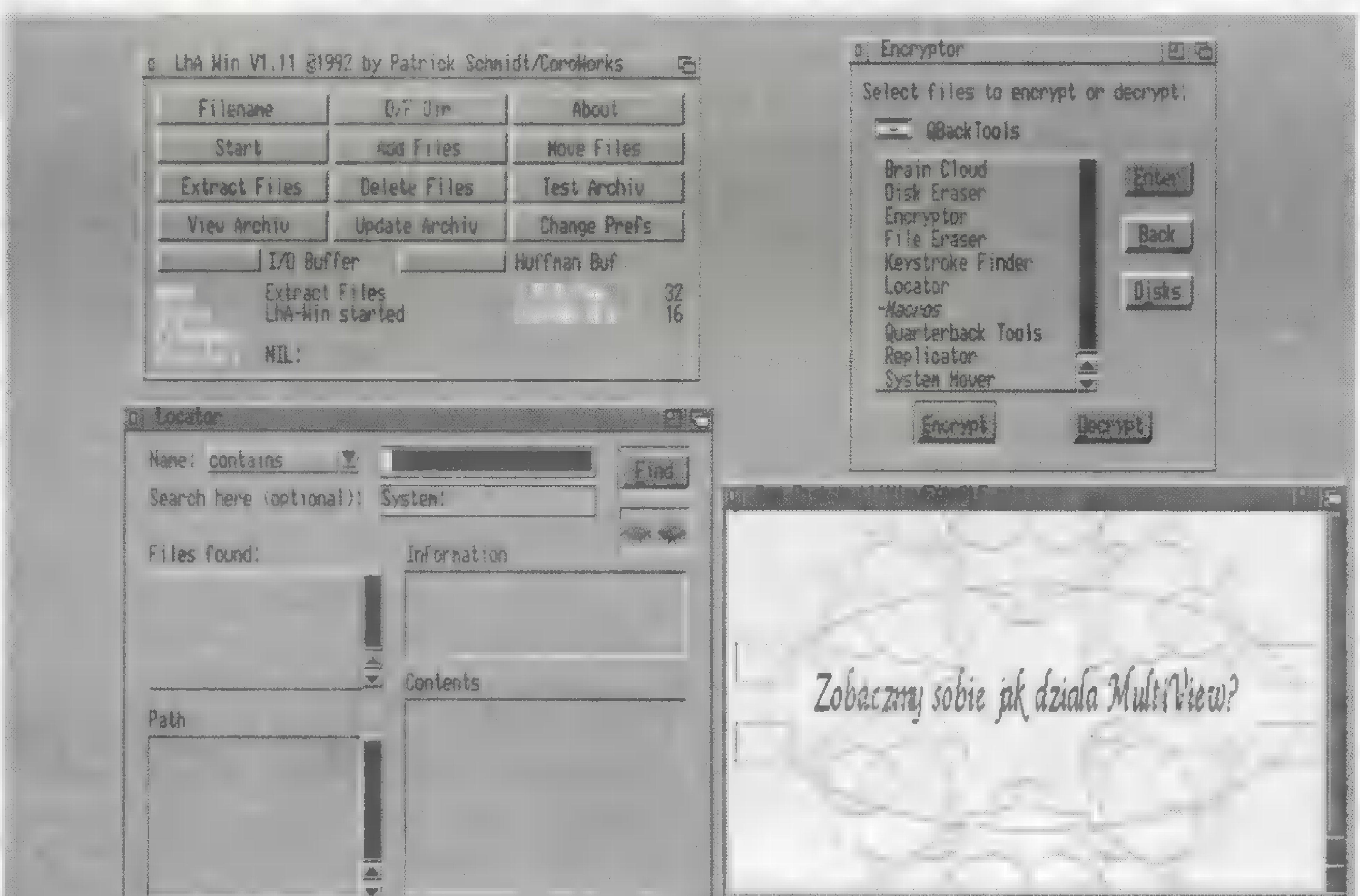
Dlaczego Opus jest "lepszy" od Workbencha?

Oczywiście nie mam zamiaru twierdzić, że Opus może zastąpić Workbench. Praca z ikonkami jest przyjemna i bardzo wydajna. W systemach powyżej 2.0 istnieje możliwość własnego urządzania "stołu pracy" (czyli Workbencha) i dzięki temu mamy bardzo szybki dostęp do najczęściej używanych programów. Często jednak zdarza się że mamy do przejrzenia różne dyski, czy też katalogi na dysku twardym. W takim przypadku praca z Workbenchem nie należy do najwygodniejszych. Owszem - pierwszy rzut oka na zawartość dyskietki można "przeprowadzić" klikając dwa razy LMB na jej ikonę i oglądając sobie poszczególne pliki, czy katalogi na niej



zawarte. Pliki/katalogi, które są niewidoczne można uwidocznić przez wybranie z menu "Windows" opcji "Show/All Files". Uruchomienie tej opcji będzie jednak wiązało się z ponownym odczytaniem zawartości dyskietki, co wcale nie musi być takie natychmiastowe. Tak naprawdę to jednak wcale nie chodzi o to, żeby zobaczyć wszystko! Gdy do naszego umysłu dociera zbyt duża ilość nieuporządkowanej informacji, nie jest on w stanie jej ogarnąć. Wprawdzie istnieje możliwość wyświetlania plików na dysku w porządku alfabetycznym. W tym celu wystarczy wybrać z menu "Windows" opcję "View by/Name". Kopiowanie plików w takim przypadku będzie odbywało się poprzez przeciąganie ich nazw z jednej szuflady do drugiej. Jest to możliwe do wykonania, ale w sumie strasznie niewygodne. Każdy kto próbował pracy z Workbenchem, a potem przesiadł się na program typu Directory Opus, nie ma wątpliwości co jest wygodniejsze.

W praktyce sprawa wygląda inaczej, gdy przeglądamy dyskietkę za pomocą Directory Opusa. Program ten wygląda zawsze tak samo, tzn. tak jak go skonfigurujemy. Dzięki temu informacje jakie do nas docierają są z założenia uporządkowane. Zawartość dysku pokazywana jest w jednym z okien w porządku alfabetycznym lub dowolnym (zależnie od ustawienia). Oczywiście szuflady wcale nie muszą być przemieszane z plikami. Poza tym istnieje możliwość uporządkowania plików pod względem daty, czy też komentarza. Wydaje się to mało istotne, ale u osób intensywnie pracujących przy komputerze są to czasami opcje absolutnie niezbędne. Directory Opus poza prostym przeglądaniem plików oferuje niesamowitą ilość dodatkowych opcji, które są dostępne tuż po naciśnięciu klawisza. Spróbujcie przeglądać teksty, czy obrazki za pomocą MultiViewera (od systemu 2.x wzwyż). Czasami się to uda, czasami będzie to jakiś zubożony kolorystycznie cień właściwego rysunku, zaś czasami Multiviewer odmówi Wam posłuszeństwa. Do tragedii dochodzi, gdy chcemy przeglądać pliki spakowane różnymi pakerami. Jeśli jest to pp.coś, czyli plik spakowany PowerPackerem, musimy uruchomić sobie PPReada, PPShowa, PPPlay'a. Posiadacze Opusa mogą uruchomić sobie PPPatchera i dalej nie martwić się o pliki spakowane PowerPackerem - zanim dotrą do Opusa zostaną rozpakowane. Gdy chcemy zobaczyć z poziomu Workbencha co znajduje się w pliku zakończonym dopiskiem ".lha" musimy albo używać Shella do ręcznego wpisywania opcji, albo działać z Lha-Winem. Program niby bardzo ładnie zrobiony, ale jednak oddzielny. Podobnie ma się sprawa z PKzipami i innymi Zoo'ami. Na dobrą sprawę wkrótce zabrakłoby nam okna Workbencha w największej rozdzielczości dla rozmieszczenia tych wszystkich "niezbędnych" programów. A co mają zrobić posiadacze gorszych monitorów, na których uzyskanie maksymalnej rozdzielczości równa się utratą wzroku w ciągu tygodnia. Suwanie Workbenchem w lewo i w prawo też nie jest chyba najwygodniejszym rozwiązaniem. Wszystkie te problemy za jednym, o pardon - dwoma, kliknięciami myszy usuwa Directory Opus. Za jednym zamachem możemy pliki przeglądać, przenosić, przezywać, przetwarzać i na dodatek w każdej chwili mamy dostęp do całej masy ulubionych programów, dla wywołania których jedynym ograniczeniem staje się ilość posiadanej w komputerze pamięci. Proszę zwrócić uwagę, że Directory Opus idzie w kierunku pełnego profesjonalizmu i 3MB pamięci okazują się wystarczające do uruchomienia wszystkich podstawowych programów (no może nie na raz, ale zawsze).



Kilka przykładów programów, które zastępuje Driectory Opus.

Directory Opus nie zastępuje Workbencha! On go uzupełnia!

Dlaczego Directory Opus jest najlepszy?

Dużo by tu wyjaśniać. Proszę po prostu przeczytać dalszą część artykułu. Chętnie zapoznam się ze wszystkimi uwagami dotyczącymi innych programów tego typu - proszę je kierować na adres redakcji. Jeśli któryś z nich będzie faktycznie lepszy, powinienem łatwo dać się do niego przekonać. Na razie jednak zostaję przy Opusie...

Dlaczego ten artykuł...

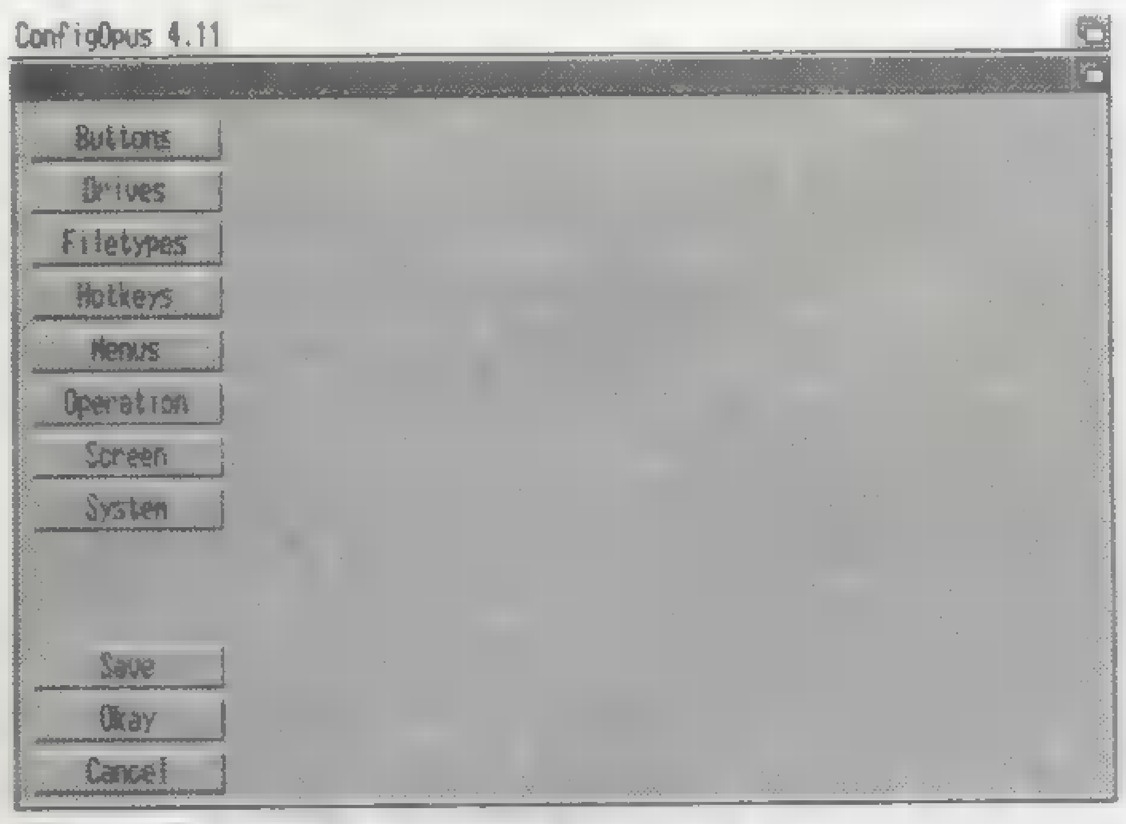
Niektórzy z Was zaczynają sobie w tym momencie stawiać pytanie: "Jaki cel ma pisanie drugiego artykułu o tym samym programie? Przecież jest on tak dobry, że się sam obroni, a te małe zmiany...". Nic bardziej mylnego!

Program jest faktycznie świetny i gdy ktoś raz go "dosiądzie", trudno mu się potem odzwyczaić. W tym jest także pewna słabość Opusa. Trzeba nieco czasu by się do niego przyzwyczaić i Ci, którzy przesiądą się na niego z innych programów tego typu, mogą mieć nieco wątpliwości. Poza tym, program jest tak dobry, że wielu z nas może się wydawać, iż nowe wersje wprowadzają tylko zmiany kosmetyczne i nie warto inwestować w ich posiadanie. Starałem się tak napisać niniejszy artykuł, żeby obalić oba powyższe mity. Program jest mimo wszystko łatwy w obsłudze, a najnowsze wersje nie tylko ją ułatwiają, ale wprowadzają wiele nowych zastosowań i udogodnień. Dążenie autorów Opusa do stworzenia dzieła doskonałego jest wspaniałym potwierdzeniem tezy, że do tej pory nie wymyślono nic takiego, czego nie dałoby się ulepszyć! Nowe Opusy są naprawdę lepsze, a

wzrost zużycia pamięci wcale nie jest proporcjonalny do wzrostu możliwości.

Najważniejsze: ustawić "robotę".

Opis nowego Directory Opusa zacznijmy od ustawienia stołu pracy, czyli "skonfigurowania" programu. Skutki tej czynności będziemy na nowo odkrywać w codziennej pracy z Opusem. Od odpowiedniego ustawienia wszystkich parametrów zależy to, jak łatwo będzie się nam współpracowało z programem. W celu wejścia do modułu konfiguracji należy wcisnąć gadżet "C" w dolnym prawym rogu ekranu lub wybrać opcję "Configure" z pierwszego menu ("Projekt").



Panel konfiguracyjny.

Mamy tutaj możliwość wybrania "zestawu", który chcemy w danej chwili konfigurować:

Buttons - jest to konfiguracja przycisków z dolnej części Opusa.

Drives - tu ustawiamy najczęściej używane ścieżki dostępu, które potem będziemy mogli wywoływać jednym kliknięciem klawisza myszy.

Filetypes - mówimy tu Opusowi jak ma się zachować w przypadku napotkania plików zakończonych odpowiednimi dopiskami typu: .pic, .mod, .lha, .zip, .sound, itp.

Hotkeys - ustalamy tu skróty, które pozwolą nam na wywołanie z klawiatury dowolnych funkcji, czy programów - wedle naszego uznania.

Menus - programujemy dodatkowe menusy Opusa.

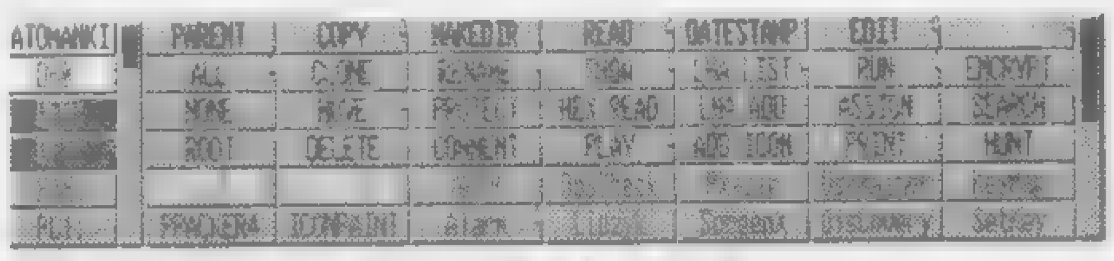
Operation - tu programujemy zachowanie Opusa przy wykonywaniu poszczególnych operacji, np. kopiowania, podwójnego kliknięcia myszką na daną nazwę, itp.

Screen - ustalamy wygląd i ustawienie szczegółów ekranu Opusa.

System - jest to punkt, w którym określamy współpracę Opusa z systemem Amigi.

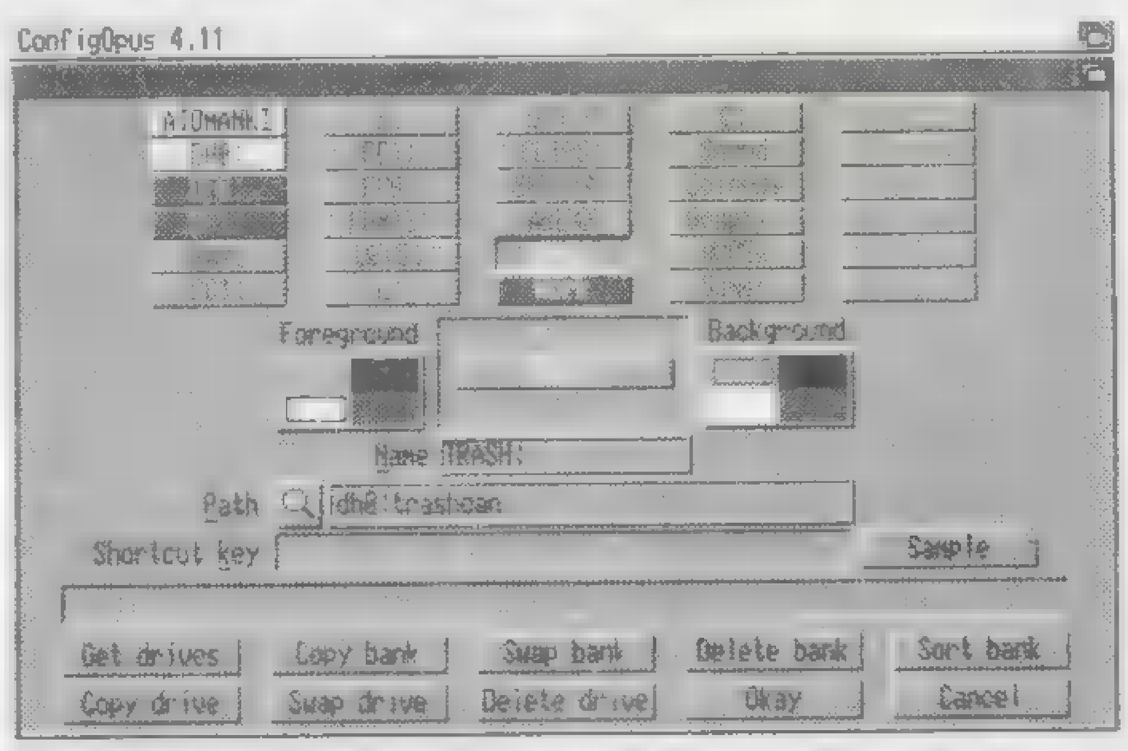
Aby wciskanie stało się działaniem...

W dolnej części ekranu Opusa mamy zbiór przeróżnego rodzaju guzików do wciskania, podzielonych na kolumienki. Guziki te nie tylko przeważnie mają podwójne funkcje, ale zostały pogrupowane tzw. bankami, które możemy wymieniać przesuwając odpowiednie suwaki.



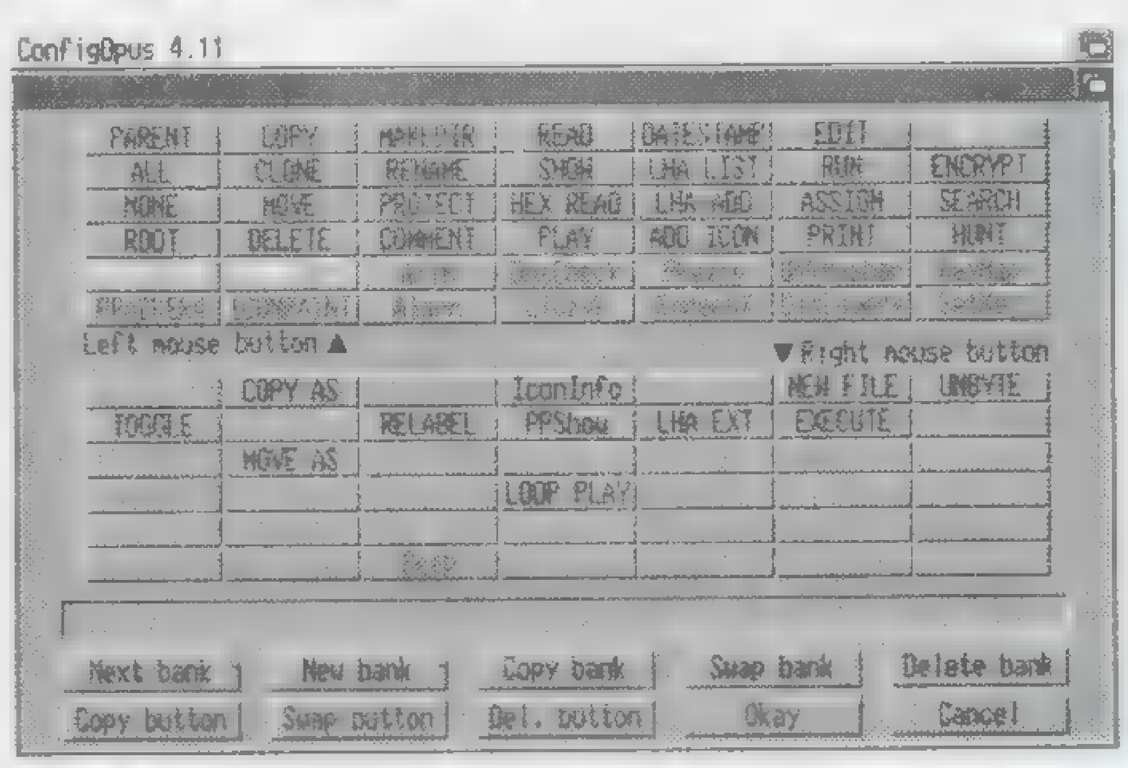
Wybieramy ścieżki i akcje.

Pierwsza z kolumienek służy do wyboru najczęściej używanych ścieżek dostępu. Innymi słowy: poprzez naciśnięcie guzika wybieramy skąd będą wczytywane katalogi. Mogą to być oczywiście stacje dysków (np. df0:, df1:, dh0:, itd.), czy też urządzenia logiczne: S:, MOJ_Dysk:, Work:, Rysunki:, Fonts:, c:, itp. Wymiana banków następuje poprzez przesunięcie suwaka lub kliknięcie prawym klawiszem myszy (RMB) nad polem z guzikami ścieżek.



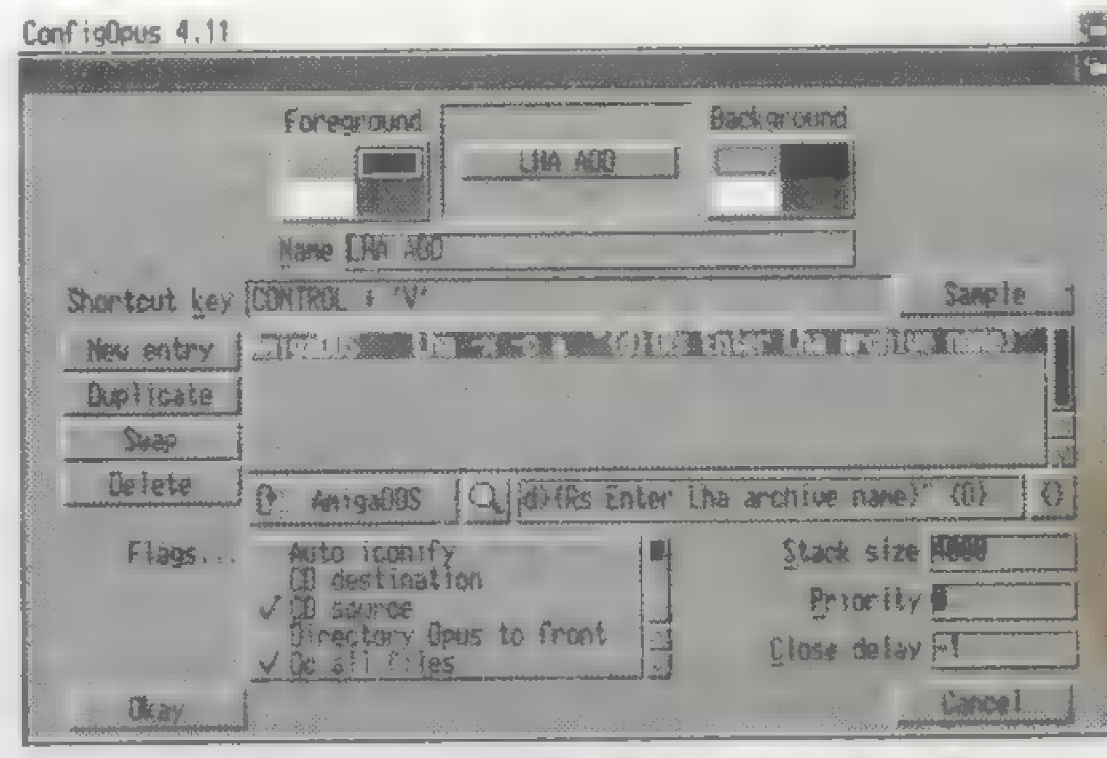
Tak wygląda "prawidłowo" sformowany przycisk dla dostępu do kosza.

Drugi zestaw guzików może być dowolnie definiowany i służy do wywoływania najczęściej stosowanych funkcji i działań. Każdy użytkownik konfiguruje go tak jak mu wygodnie lub jak się przyzwyczaił. Zaznaczam, że do dobrego tonu należy NIE krytykować ustawienia tych guzików u kolegi, bo po to jest Opus, żeby każdy miał prawo pracować sobie tak jak chce. W końcu mamy demokrację!



Wybieramy sobie guzik...

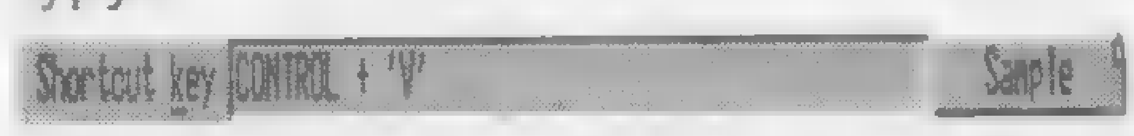
Zajmijmy się poszczególnymi opcjami jakie można zmienić przy definiowaniu wywołania funkcji.



Wybraliśmy guzik Lharka (pakowacza).

Na samej górze definiujemy nazwę (Name) i kolory tła oraz czcionki w jakich nazwa będzie wyświetlana. Ja z uwagi na monitor monochromatyczny pracuję z czterema kolorami, ale nic nie stoi na przeszkodzie, żeby sobie poużywać do 16-tu kolorów (jeśli ktoś lubi bić się po oczach...). Poniżej możemy zdefiniować dowolne wywołanie z klawiatury danej funkcji. Wciśnięcie gadżetu "Sample" umożliwi nam wręcz wciśnięcie dowolnej kombinacji klawiszy, która zostanie zapisana w okienku obok.

Poniżej mamy panel definiowania funkcji. Funkcje mogą mieć następujące typy:



Command - wywołanie wewnętrznej funkcji Opusa.

AmigaDOS - najzwyklejsze w świecie wywołanie odpowiedniego programu.

Workbench - podobnie, ale z określeniem ścieżki dostępu jako SYS:

Batch - wywołanie tekstu zbudowanego tak jak programiki CLI (coś w rodzaju Startup-Sequence)

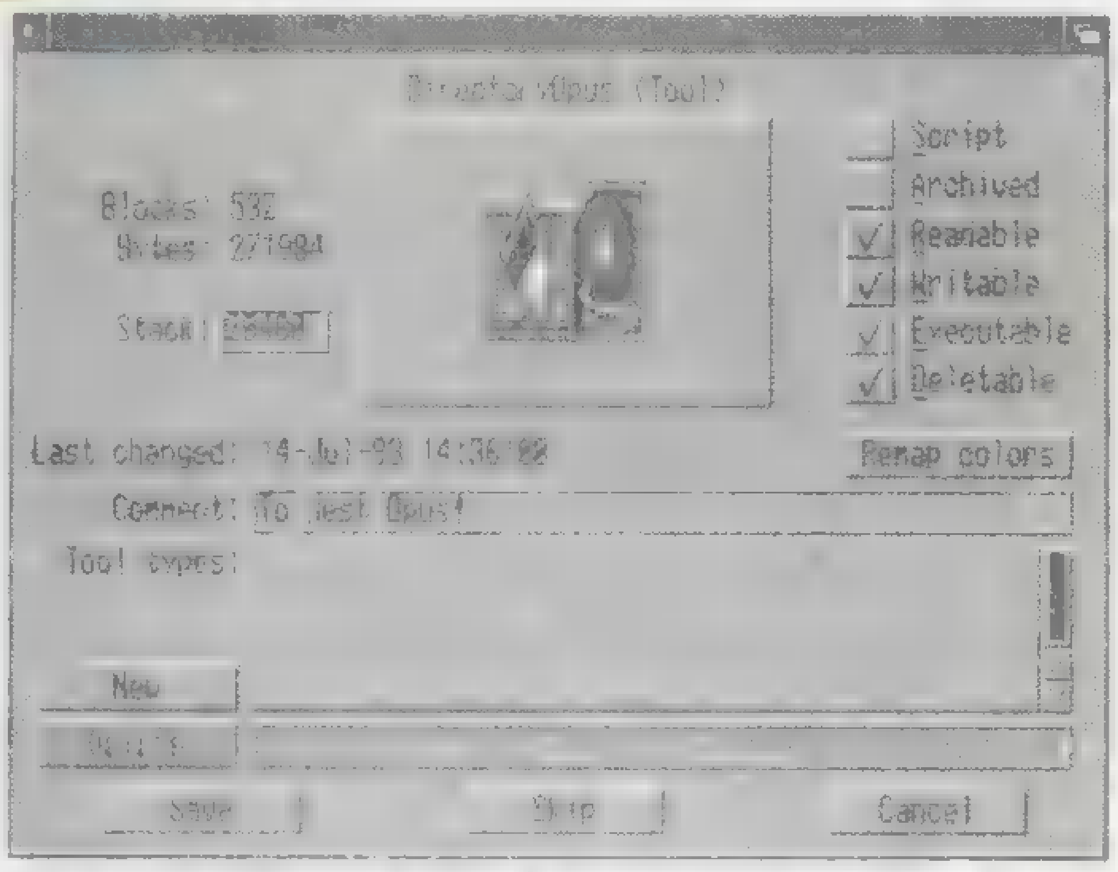
AREXX - wywołanie programiku w ARexxie.

Najwięcej kłopotu przysporzy Wam oczywiście zapoznanie się z wewnętrznymi funkcjami Opusa. Aby ułatwić Wam orientację pozwoliliśmy sobie na zastosowanie triku graficznego i przedstawienie wszystkich tych funkcji na jednym rysunku. Funkcje te są właśnie podstawą niesamowitych możliwości Opusa.

Warto tu zwrócić uwagę, że te funkcje, które mają dopisek "BG" (BackGround) mogą działać w tle, bez zajmowania Opusa, jak to było w jego wcześniejszych wersjach. Oczywiście pełne wykorzystanie funkcji możliwe jest faktycznie dopiero przy użyciu programików w ARexxie. W takim przypadku nasz programik może przejąć pełne sterowanie Opusem i łączenie

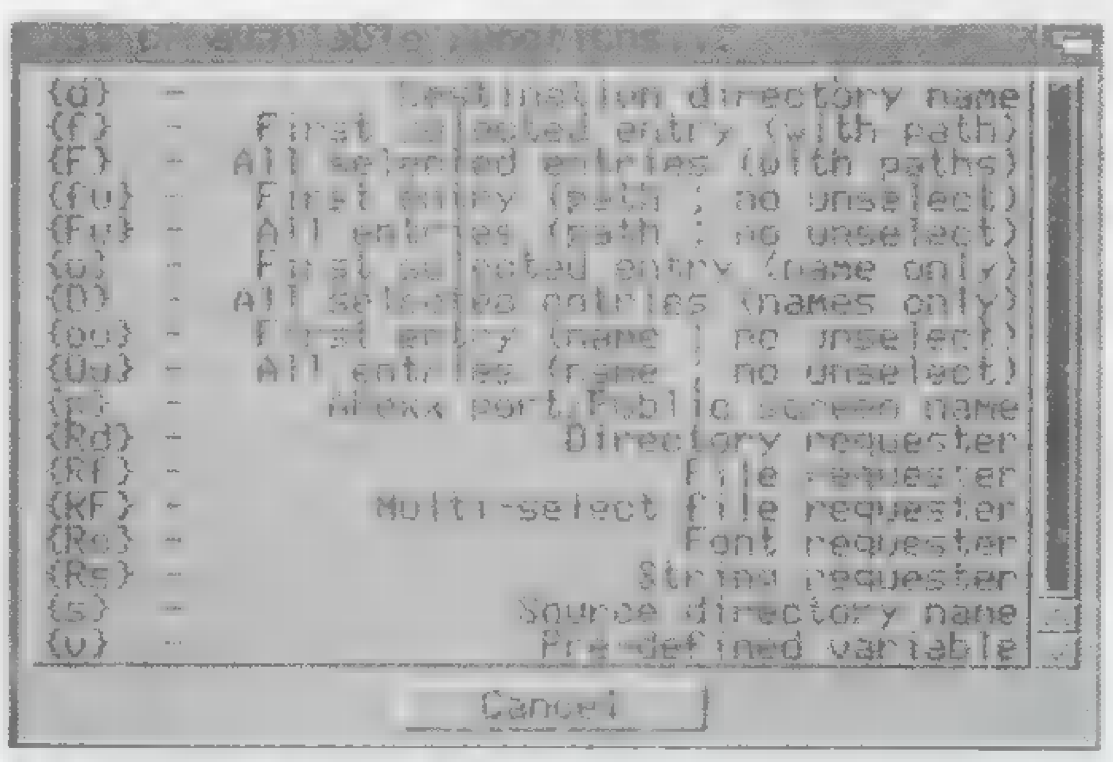
jego funkcji z funkcjami innych programów.

Jedną z ciekawszych, nowych funkcji, która zastępuje kawałek Workbencha jest "IconInfo". Przy jej pomocy możemy sobie wywołać informacje tak jak z menu "Icons/Information..." i dostosować ją do własnych potrzeb.



Nareszcie cała informacja pod ręką.

Funkcje współdziałają oczywiście z tym, co jest wyświetlone w oknach Opusa. Dzieje się to poprzez wstawienie do nich (a także do wywołań dowolnych programów) odpowiednich argumentów. Wywołanie wszystkich możliwych argumentów i wybranie jednego z nich umożliwia nam przycisk "{}".



Lista argumentów.

Argumentami mogą być zarówno zaznaczone pliki, jak i odpowiednie requestery (okna), które pozwolą nam na wczytanie, czy to nowej nazwy, czy to odpowiedniej czcionki, itp.

Przykładowe wywołanie funkcji pakowania programem Lharc wygląda następująco:

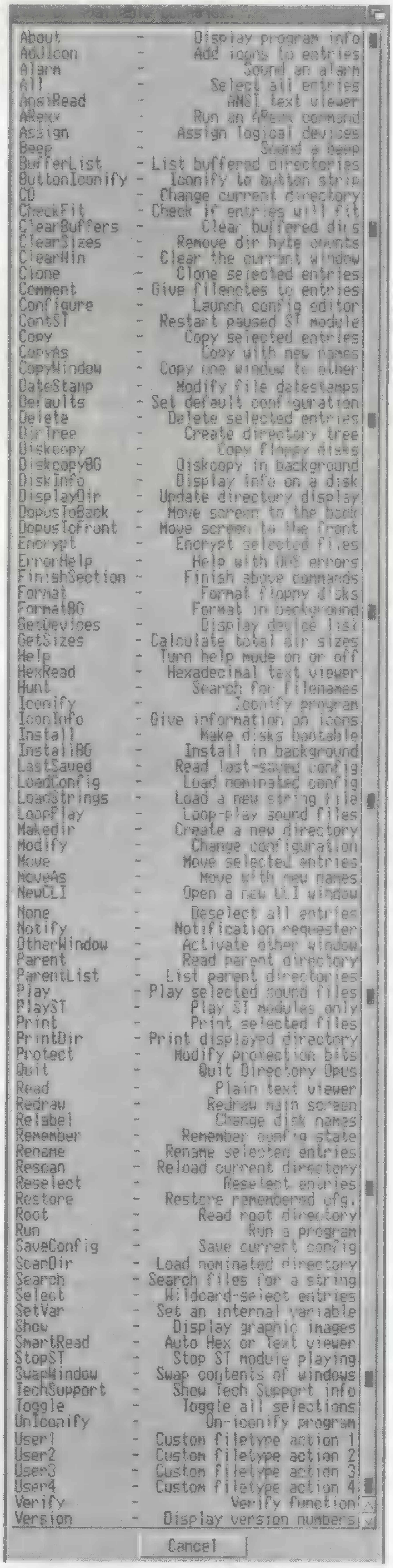
c:Lha -x -c a "{d}{RS Enter Lha archive name}" {O}

gdzie:

c:Lha - wywołanie programu Lha z katalogu c:

-x -c a - argumenty programu mówiące o tym, że dany plik/pliki będą pakowane łącznie ze ścieżkami i to co zostało spakowane zostanie sprawdzone.

"{d}{{RS Enter Lha archive name}" - ścieżka docelowa połączona z wpisaną do specjalnego gadżetu nazwą.



Wykaz wewnętrznych funkcji Opusa.

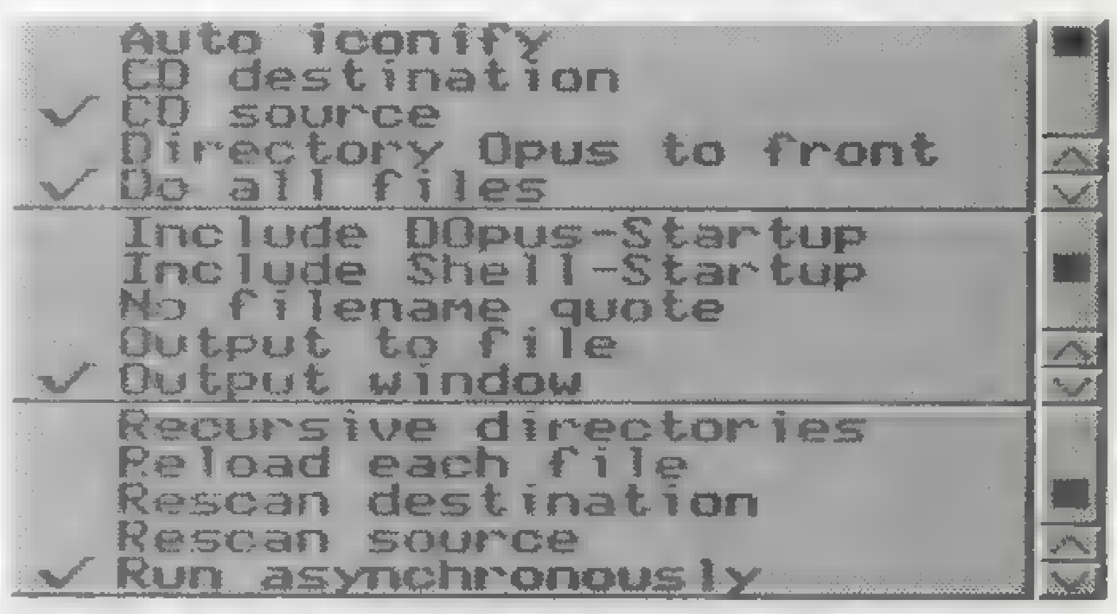
{O} - nazwy wszystkich zaznaczonych plików.



Tu wrzucasz ikonki programów, które chcesz uruchamiać z Opusa.

Twórcy Opusa do maksimum udogodnili wpisywanie programów "pod przyciski". Wystarczy na ekranie Workbencha chwycić ikonkę programu i wrzucić ją na odpowiednią ikonkę Opusa, która się pojawia w trakcie dopisywania nowych opcji.

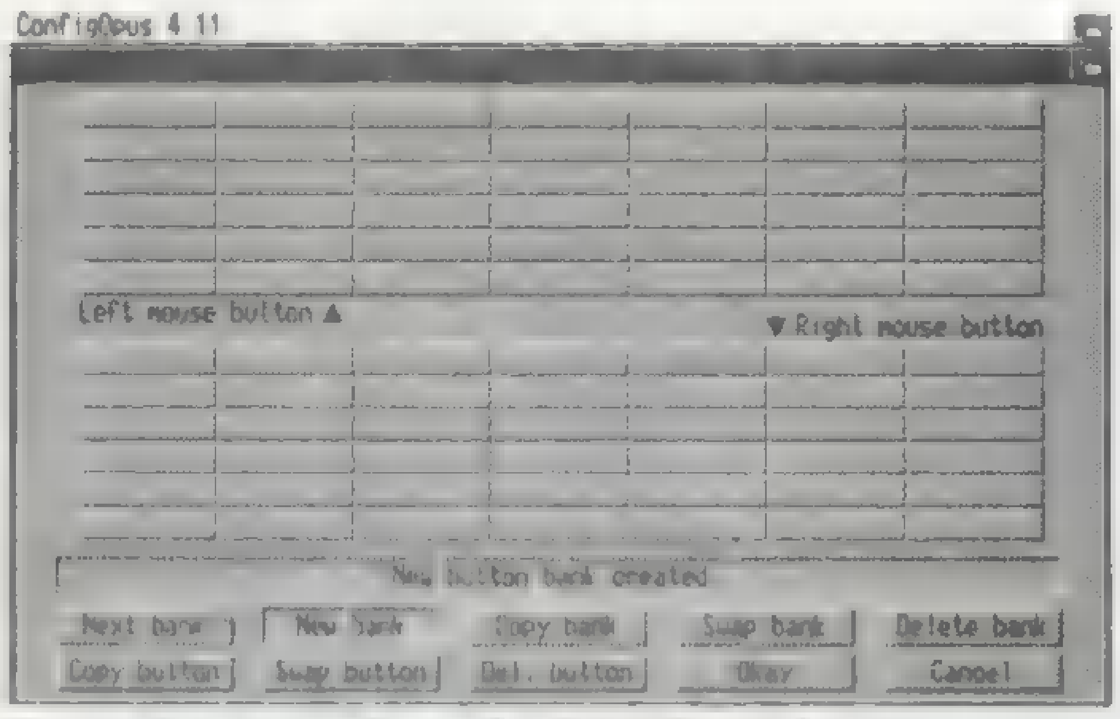
W dolnej części okna służącego do określania wywoływanej funkcji ustalamy wielkość stosu jaka ma być zarezerwowana przy uruchomieniu funkcji, jej priorytet (normalnie 0), sposób zakończenia funkcji (Close Delay - przy Lharcu polecam -1, co równa się konieczności wciśnięcia LMB) oraz flagi. Flagi są to dodatkowe parametry, które uskuteczniają działanie programu.



To Ty ustalasz jak działa Twój program.

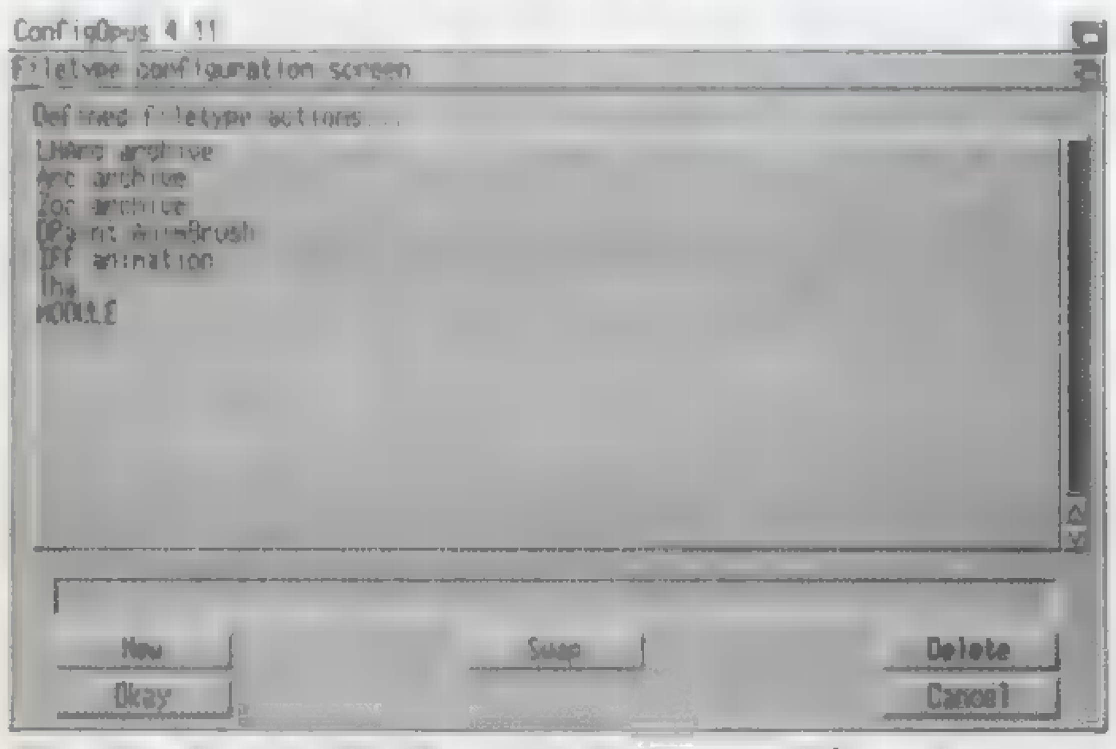
Dzięki nim możemy ustalić, czy w trakcie wykonywania danej funkcji Opus ma zostać zwinięty (zikonifikowany - Auto iconify) do cienkiej belki na ekranie Workbencha, czy też ma tylko zostać wypchnięty do tyłu (Workbench to front). Dzięki flagom możemy też ustalić, czy operacje będą wykonywane na wszystkich plikach (Do all files), czy tylko na pierwszym z brzegu. One też pozwalają nam na zwolnienie Opusa tak, że dana czynność będzie wykonywana na osobnym tasku (Run asynchronously). W praktyce ustawienie flag zostaje wielokrotnie zweryfikowane przez życie...

Po ustaleniu wszystkich parametrów naciskamy klawisz "Okay", co powoduje przejście do panelu wyboru guzików i zarządzania poszczególnymi ich bankami (zestawami). Banki można dodawać, kasować, czy też przemieszczać. Podobne operacje wykonuje się na poszczególnych guzikach (Buttons).



Panel zarządzania bankami.

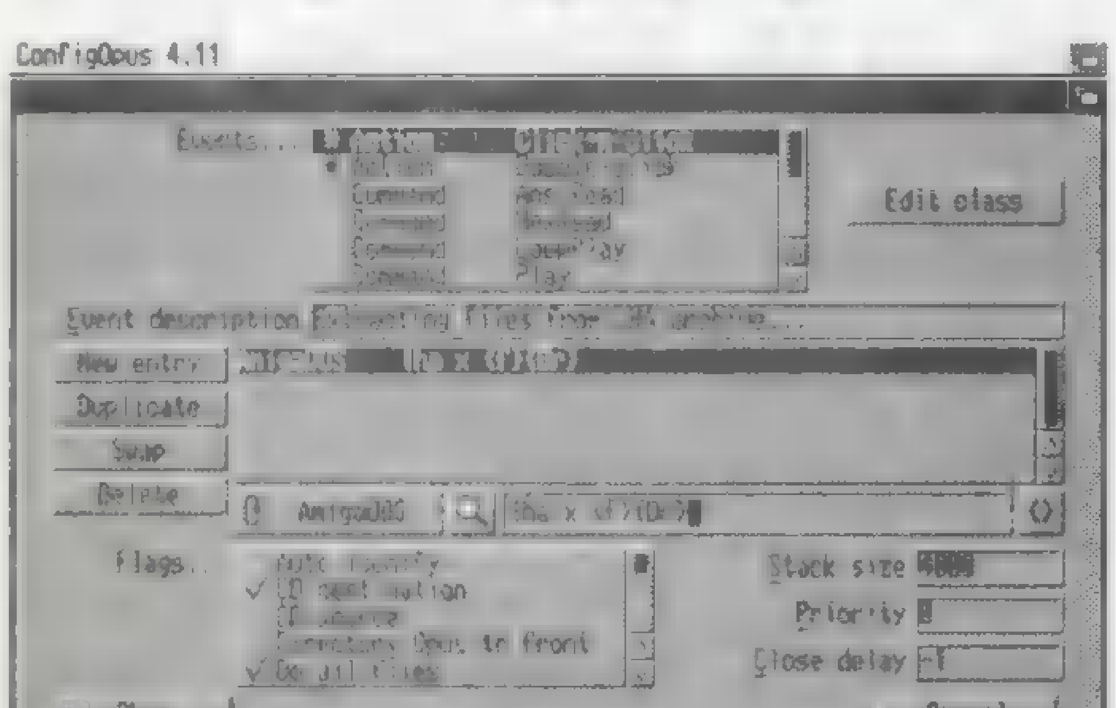
Typologia stosowana.
Następną pozycją w liście konfiguracyjnej są typy plików (Filetypes). Po wciśnięciu tego przycisku pojawia nam się okno wyboru typu pliku.



Rodzaje plików zależą głównie od przyrostka, ale nie tylko...

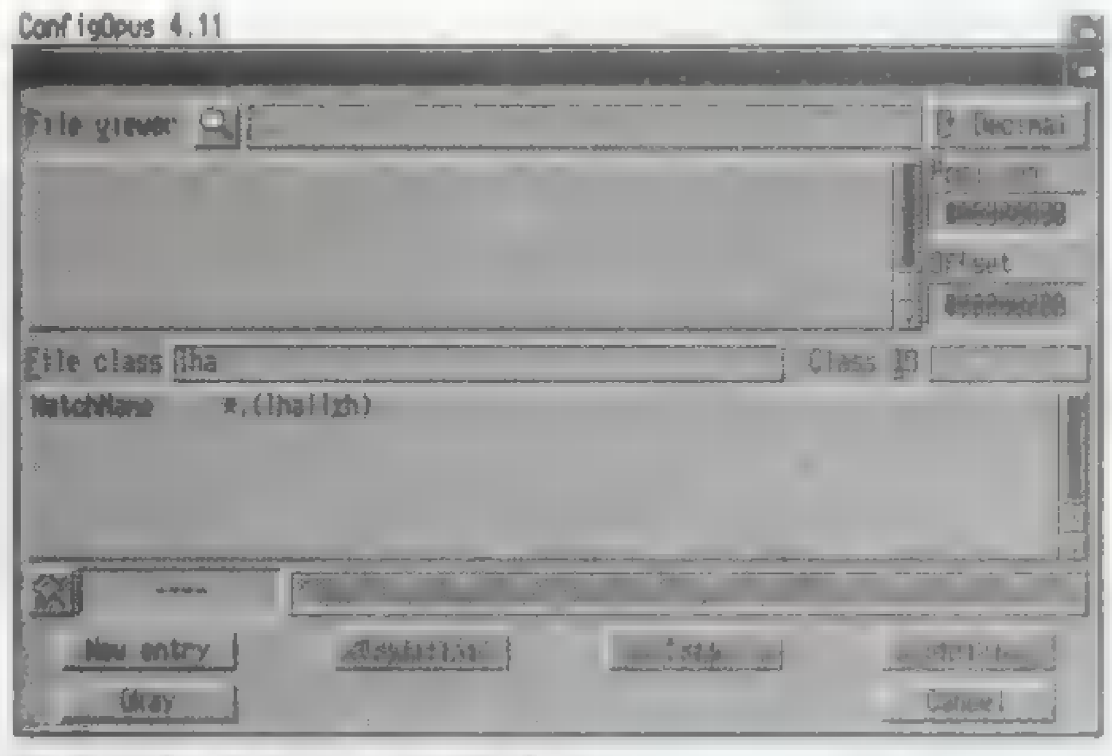
Opcja ta pozwala nam na określenie zachowania się Opusa po dwukrotnym kliknięciu LMB nad nazwą z określonym dopiskiem. Np. jeżeli w oknie Opusa będziemy mieli coś w stylu "nazwa.lha", to możemy sobie zdefiniować, że w przypadku podwójnego kliknięcia na tę nazwę program odczyta nam zawartość spakowanego pliku. W innym przypadku może to być uruchomienie jakiegoś nietypowego programu muzycznego i odegranie zaznaczonego pliku. Przykładów może być mnóstwo.

Okno ustalania parametrów, które będą wykorzystane w danym działaniu wygląda podobnie jak przy definiowaniu znaczenia poszczególnych guzików.



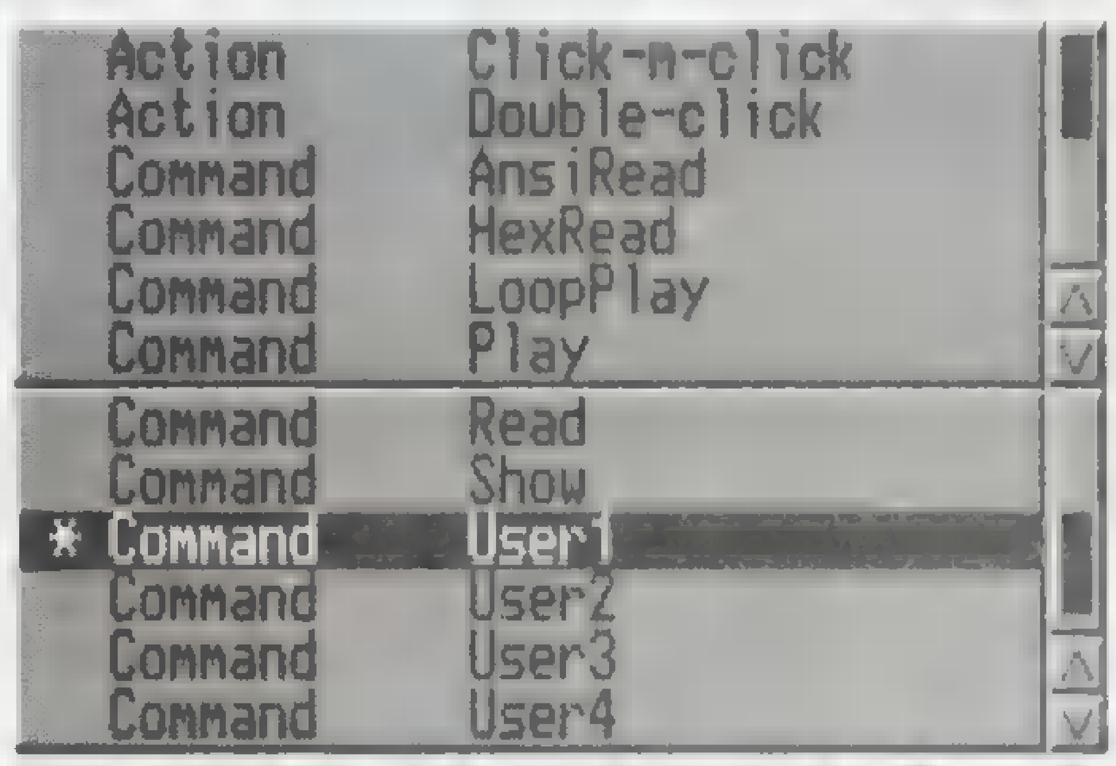
Okno określania akcji dla plików zakończonych na ".lha"

W ten prosty sposób możemy ustalić choćby kilkanaście różnych końcówek, jakie będzie rozpoznawał Opus



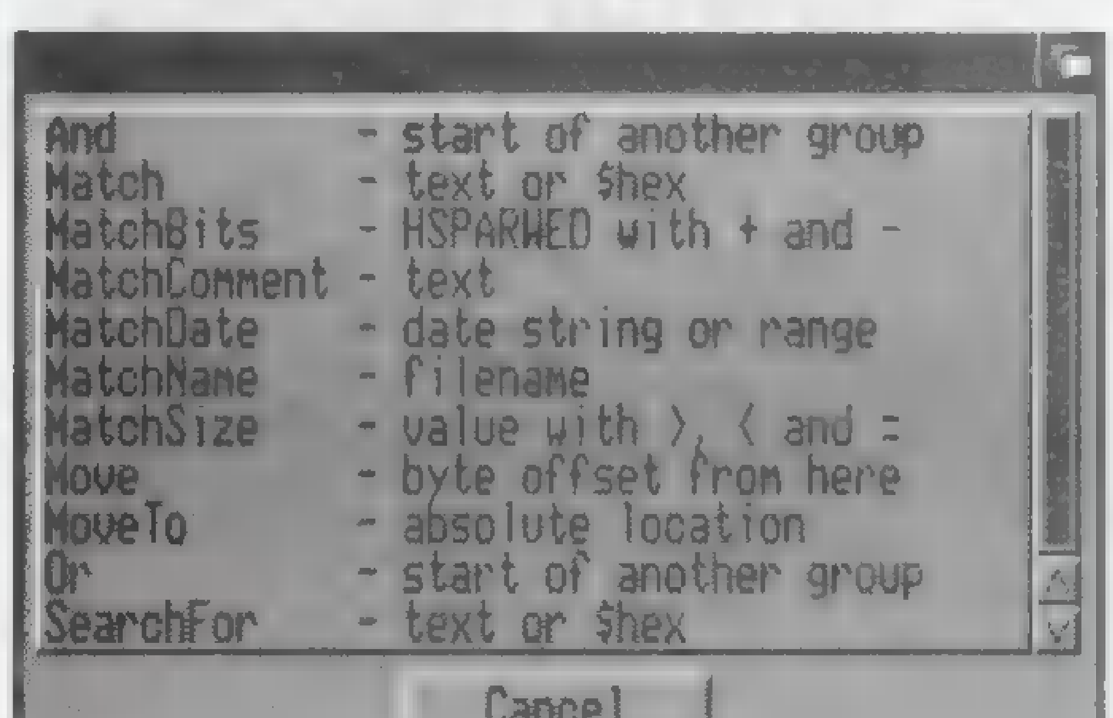
Edycja klasy plików.

i jak będzie na nie reagował. Rodzaj reakcji może być uzależniony od tego jak potraktujemy dany plik myszą. Czego innego możemy wymagać od Opusa gdy klikniemy na niego dwa razy, czego innego zaś gdy raz na niego, a za drugim razem na przeciwne okno.



Kto zgadnie, jak zachowa się Opus?

Pliki możemy wybierać pod względem różnych kryteriów. Najczęstszym był do tej pory (w poprzednich Opusach) oczywiście dopisek po kropce, ale teraz może to być zgodność daty, wielkości, komentarza, umiejscowienie pliku, itp. My podajemy schemat wyboru plików, a Opus załatwia za nas automatycznie resztę.



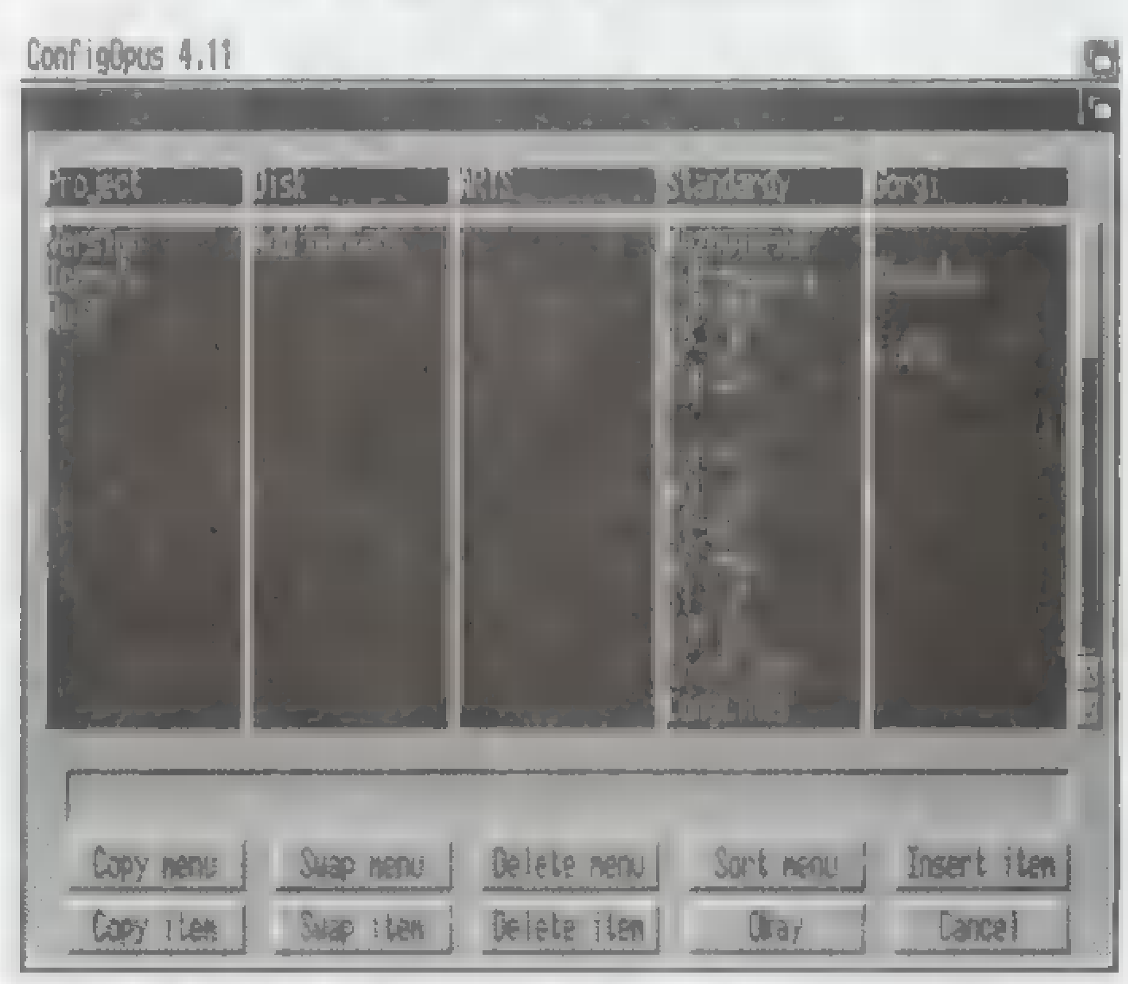
Pliki określamy według tych cech.

Wybranie akcji jaka ma odpowiadać danej funkcji przy danym pliku następuje poprzez wpisanie "zadania" poniżej rubryki określającej działanie (Event description). Potwierdzenie poprzez "Okay" spowoduje, że gdy następnym razem wejdziemy do danego rodzaju pliku, przy danej funkcji pojawi się gwiazdka.

Gorące klucze...
Podobnie wygląda definiowanie działania wywołanego określoną kombinacją klawiszy: Hotkeys. Wszelkim kombinacjom klawiszy możemy przypisać wywołanie dowolnych programów. Nie jest tu nawet istotne, czy w momencie wciskania klawiszy Opus będzie akurat aktywny z przodu, czy też nieaktywny - zikonifikowany: skrót i tak zadziała. Przy dużej ilości pamięci wystarczy w takim układzie odpalić Opusa, żeby mieć dostęp do innych, częściej używanych programów z klawiatury.

Kucharz serwuje.
Zupełnie podobnie jak klawisze definiuje się opcje w menu.

Istnieje tu nawet możliwość zmiany "oryginalnych" komend zamontowanych tam przez twórców Opusa - wystarczy kliknąć nad nazwą funkcji, którą chcemy zmienić i już pojawi się przed nami jej wywołanie. Menu są szczególnie przydatne przy wywoływaniu funkcji, które rzadziej używamy. Ja np. w jednym menu zmagazynowałem sobie wszystkie mini programiki do tłumaczenia "standardów" polskich znaków. Są to jakby prościutkie panele, które dzięki Opusowi nie potrzebują specjalnego okna, czy systemu obsługi jaki kiedyś próbowałem zastosować w Textonie. Teraz wystarczy zaznaczyć pliki w jednym oknie, w drugim wczytać ścieżkę do której plik ma zostać przeniesiony i wybrać odpowiedni system tłumaczenia "z" "na" z menu.



Panel do wybierania menu.

Na dzisiaj skończymy definiowanie Opusa. Pozostałymi opcjami oraz krótkim wprowadzeniem w działanie Opusa zajmiemy się w następnym AMIGOWCU.

Wszelkie uwagi dotyczące poruszanego tematu proszę kierować na adres redakcji. ❑

PROtracker

Marcin Dachowski

Protracker w najbardziej rozpowszechnionych wersjach (1.1 i 2.3) jest dla mnie najlepszym programem jaki kiedykolwiek napisano na Amigę. Mimo paru błędów, pracuje się z nim po prostu miło i wygodnie. Od nowej wersji tego programu oczekiwałem jedynie, by nie była gorsza od już istniejących. Niestety, Protracker 3.01 okazał się przykładem przekombinowania. Przykładem na to, że przy poprawianiu jednego błędu, można zrobić dziesięć innych.

Po uruchomieniu programu moim oczom ukazał się obraz nasuwający skojarzenia z pulpitem samolotu. Nieczytelne opisy gadgetów, dwa razy węższy obszar edycji, nowe rozmieszczenie podstawowych funkcji. To wszystko ma mi ułatwić pracę? Tryb Med-Res (640x200) przy trzech bitplanach dość skutecznie męczy wzrok po parogodzinnej pracy. A zatem, po co? Przecież bardziej zagmatwany, a przez to bardziej profesjonalny wygląd nie oznacza, że program jest bardziej przydatny niż jego poprzednie wersje. Tym



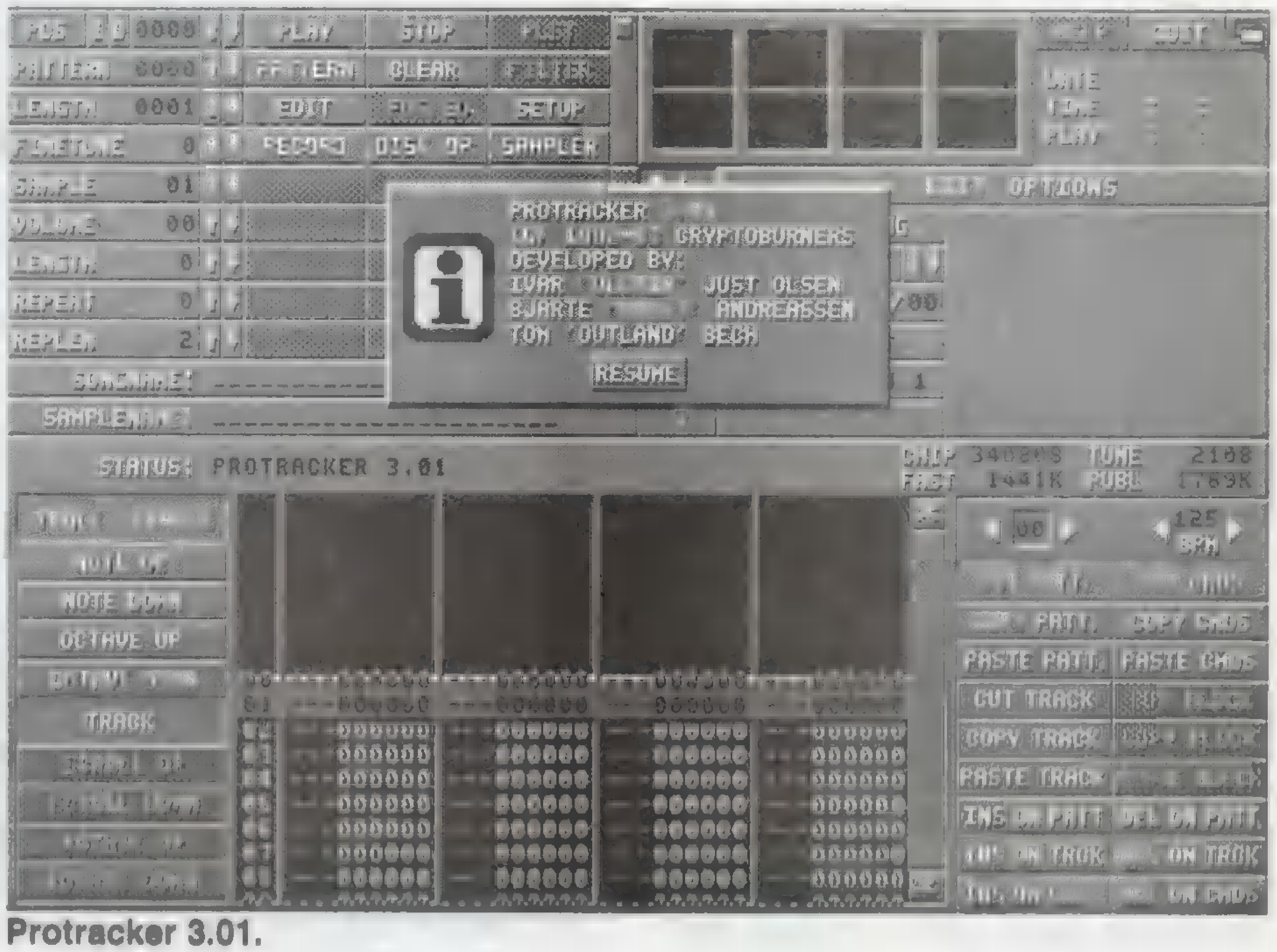
Protracker 3.01.

bardziej, gdy podczas tworzenia owego bardziej profesjonalnego wizerunku zapomina się o wielu podstawowych funkcjach i popełnia się masę błędów.

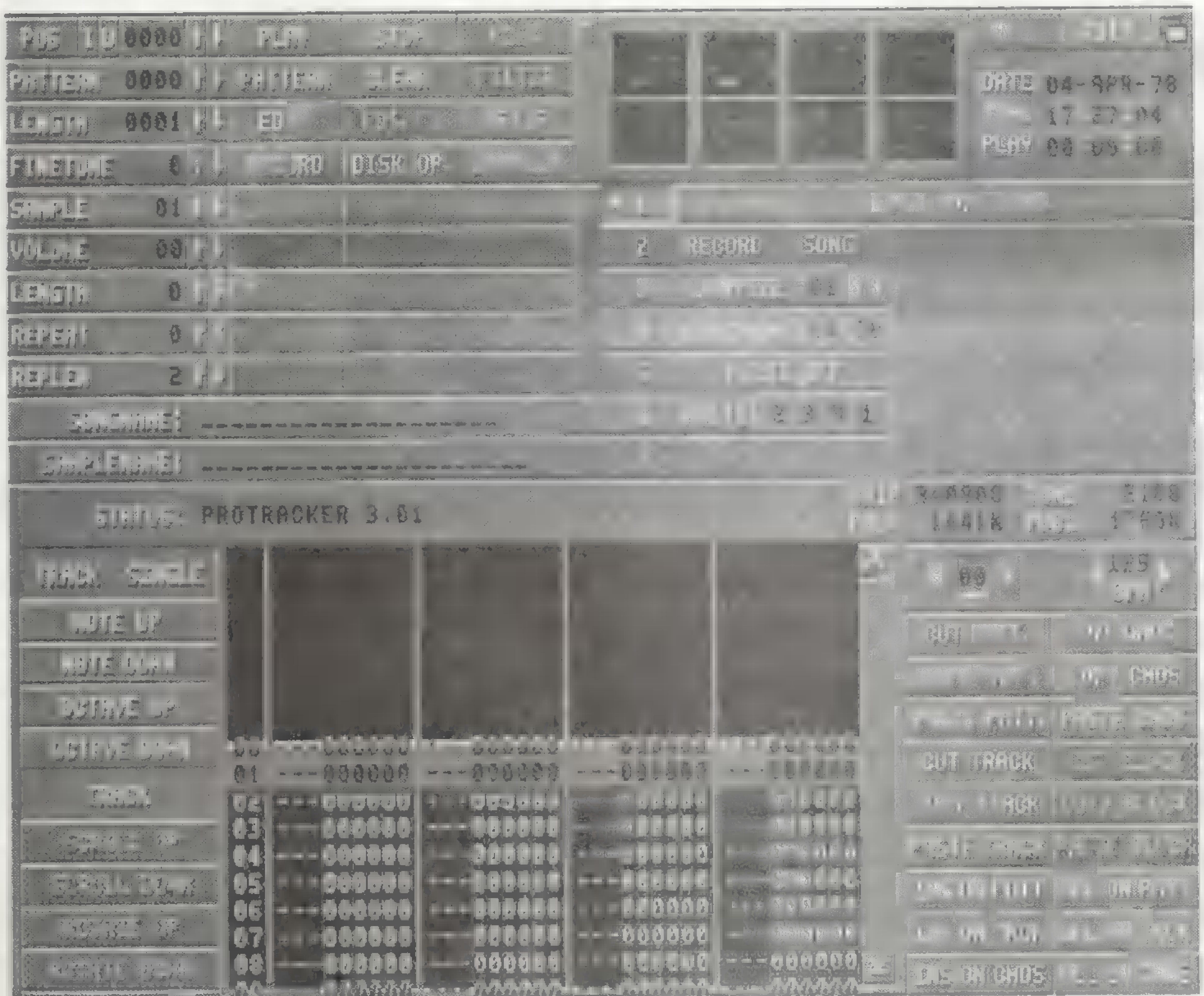
Najpoważniejszą wadą, w nowym Protrackerze, jest niczym nie uzasadnione, nie akceptowanie niektórych komend i złe odtwarzanie utworów. Ciekawe, że dzieje się to w najmniej oczekiwanych momentach (prawa Murphy'ego?!) i kończy zazwyczaj zawieszeniem systemu. Protracker jako program używany do tworzenia muzyki lub chociażby jej słuchania nie przedstawia większej wartości. Pozostaje jednak niezłym narzędziem do obróbki sampli, ale o tym za chwilę.

Innymi istotniejszymi wadami są: brak operacji na blokach ("ślepe" gadżety), niemożność zgrywania poszczególnych patternów, brak automatycznego tworzenia podstawowych akordów (durowych, mollowych i septymowych, tak jak w wersji 2.3), zupełnie nie przemyślany blok odpowiedzialny za współpracę z dyskiem, brak edytora PLST oraz rezygnacja z MIDI (chociaż niektórzy kwalifikują to jako zaletę).

Z najbardziej przydatnych opcji wyróżniłbym nowy CHORD EDITOR (mimo opisanej wyżej wady), który pozwala



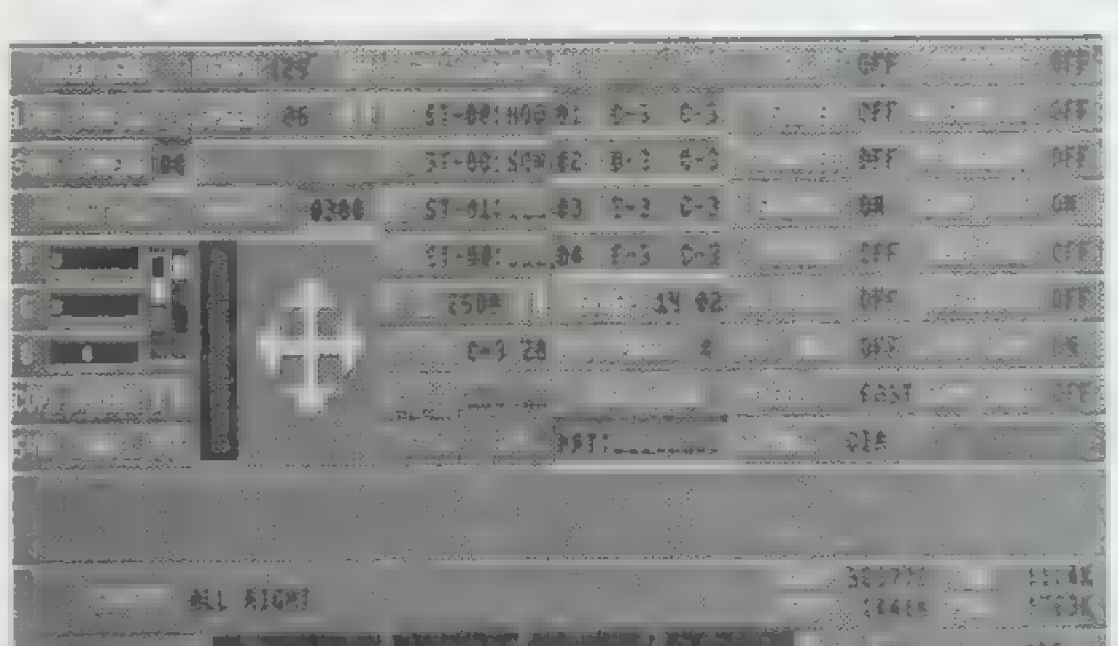
Protracker 3.01.



Protracker 3.01 - ekran edycji.

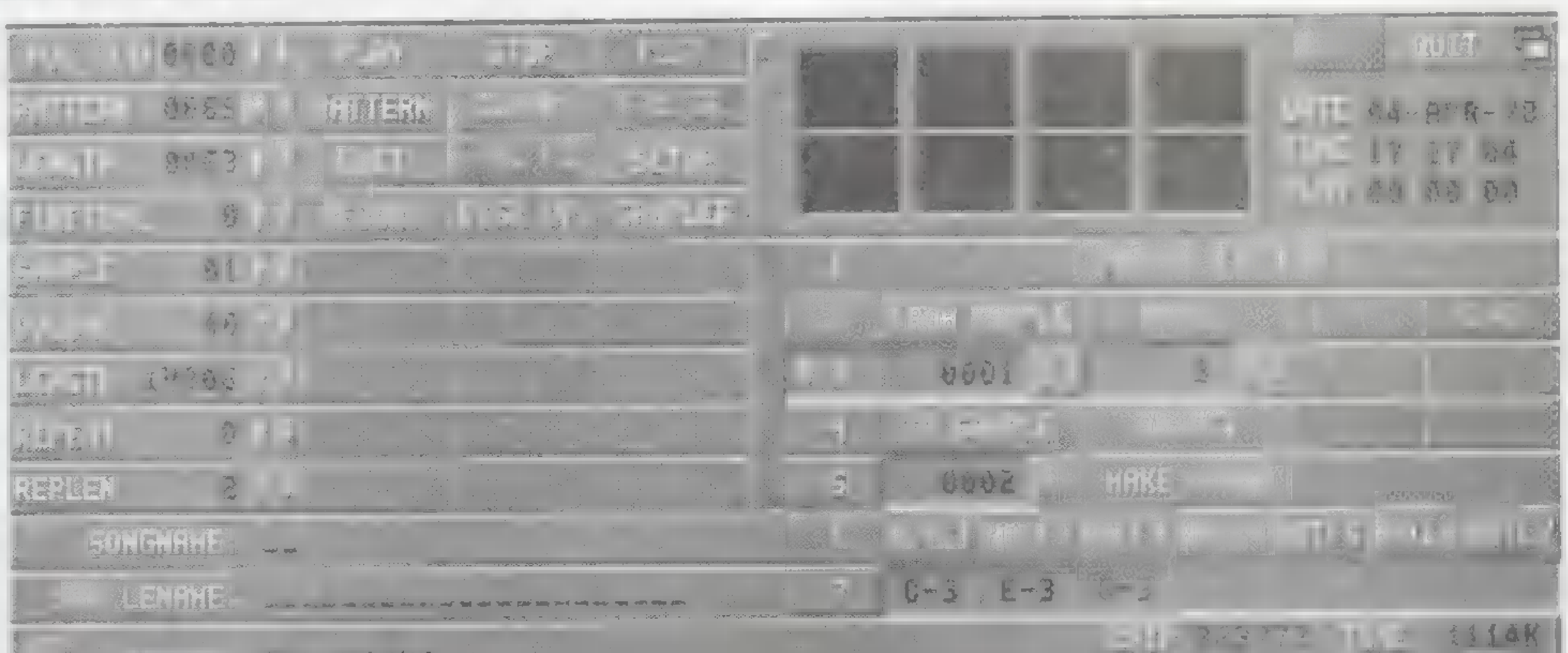
na tworzenie, nawet bardzo rozbudowanych, akordów (do siedmiu dźwięków). Posiada także niezwykle przydatną opcję ADJUST, dzięki której stworzony akord ma taką samą głośność jak dźwięk bazowy i w miarę dobrą jakość. Nowy CHORD EDITOR jest dla mnie jedyną przydatną funkcją w Protrackerze 3.01.

Nowościami w programie są także: nieco szybszy sampler (szybszy przy resamplowaniu, zmianie głośności, itp.), wreszcie poprawiona opcja formatowania dysku (ze "śmieciami" na ekranie,



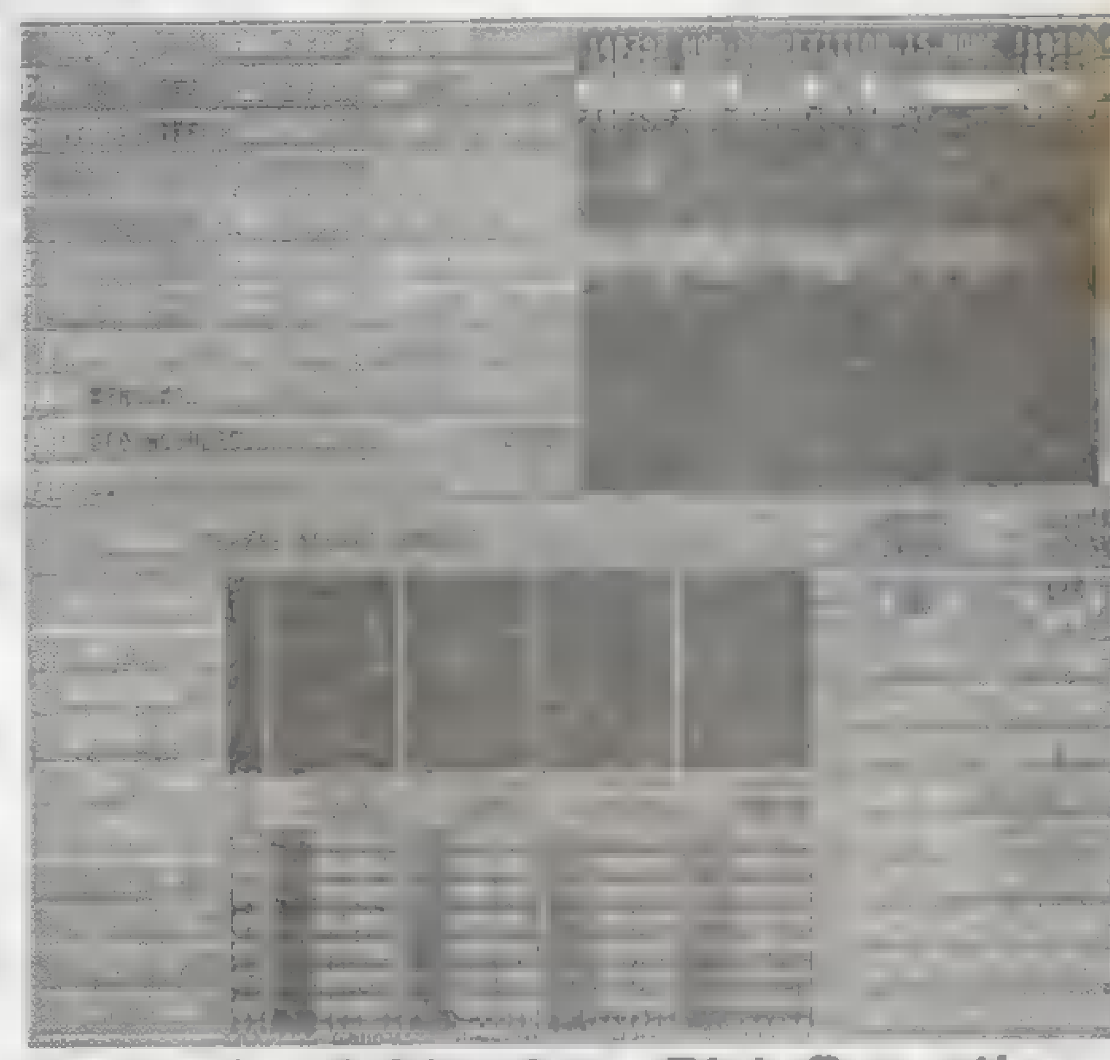
Protracker 3.01 - ekran Setup.

ale lepsze to niż nic), sterowane gadżetami operacje na patternach, ścieżkach i komendach (dotychczas dostępne tylko z klawiatury), podawane dziesiętnie długości sampli i ich zapętleń oraz nowe, ostrzegające requestery, przy operacjach bezpowrotnie zmieniających charakterystykę sampla (np. Upsample i Downsample).



Protracker 3.01 - sampler + edytor akordów.

W obecnej chwili nikomu nie polecam Protrackera 3.01 do poważnej pracy. Mimo wielu nowych rozwiązań, program posiada wady, które dyskwalfikują go w oczach każdego muzyka.



Protracker 3.01 - ekran Disk Operation.

Nie przeczę jednak, że po wielu koniecznych zmianach, a także po dołączeniu zapowiadanych nowych opcji (jak np. całkiem nowy sampler, nagrywanie plików jako executable, czy też rozszerzenie długości sampla z 64 do 128 KB), program ten stanie się bardziej użyteczny niż wersja 2.3 i mimo zabójczej dla oczu rozdzielczości znajdzie swoich zwolenników. Szczerze tego życzę grupie Cryptoburners, która to uraczyła nas jak na razie, niezbyt smacznym kąskiem, jakim okazał się Protracker 3.01. ❑

BRILLIANCE

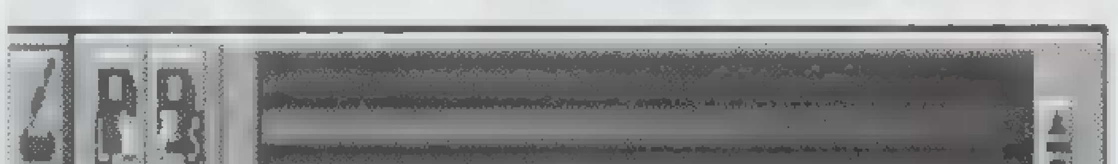
Profesjonalny program graficzny cz.2

Tomasz Łabuda

W pierwszej części opisu Brilliancea przybliżyłem Czytelnikom podstawowe opcje graficzne. Stanowią one jedynie wycinek całkowitych możliwości programu. Aktualnie zajmę się nieco bardziej zaawansowanymi technikami rysunkowymi. Ich odpowiednie wykorzystanie i ostateczny efekt, zależą jedynie od inwencji twórczej użytkownika programu.

"Brilliance" należy do programów pozwalających na stosunkowo łatwe tworzenie, nawet bardzo skomplikowanych kompozycji. Najprostszym z dostępnych efektów jest cieniowanie. Opcje z tym związane znajdują się w menu "Draw mode", do którego przechodzimy po naciśnięciu prawego przycisku myszy (przyciśnięcie lewego klawisza myszy, uaktywnia wcześniej wybraną opcję). Panel, który się pojawi, jest podzielony na szereg "okienek" (Rysunek 1). Cieniowanie kryje się w ostatnim z nich ("Gradient fill mode"). Program udostępnia nam następujące możliwości:

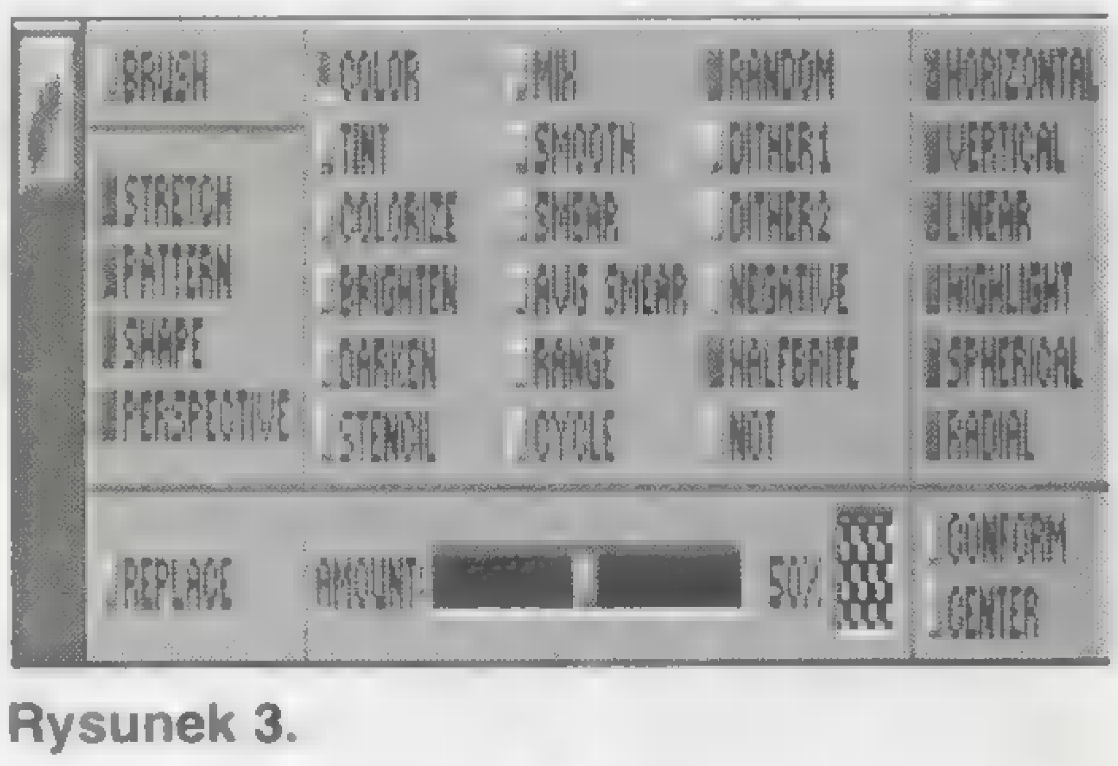
- Horizontal - poziomy kierunek przejścia kolorów,
- Vertical - pionowy kierunek przejścia kolorów,
- Linear - przejście kolorów będzie się odbywało liniowo, pod określonym przez nas kątem,
- Highlight - pozwala na bezpośrednie uzyskanie efektu oświetlonej powierzchni,
- Spherical - cieniowanie kuliste,
- Radial - cieniowanie promieniowe (koncentryczne).



Rysunek 2.

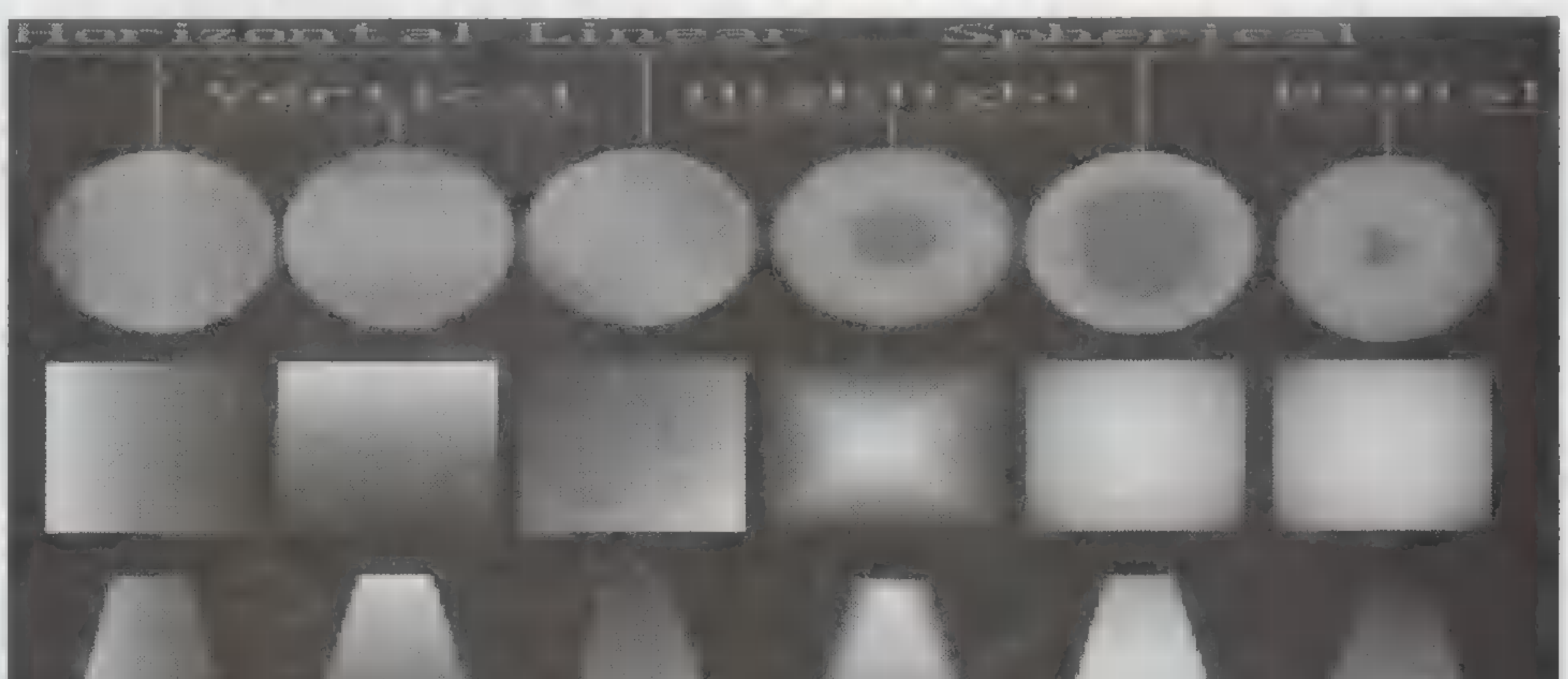
Różnice pomiędzy poszczególnymi rodzajami cieniowania pokazane zostały na rysunku ("Cieniowanie").

Chcąc uzyskać płynne przejścia kolorów należy odpowiednio przygotować paletę kolorów. Dokonamy tego w "Palette menu" (Rysunek 2) zawierającym:

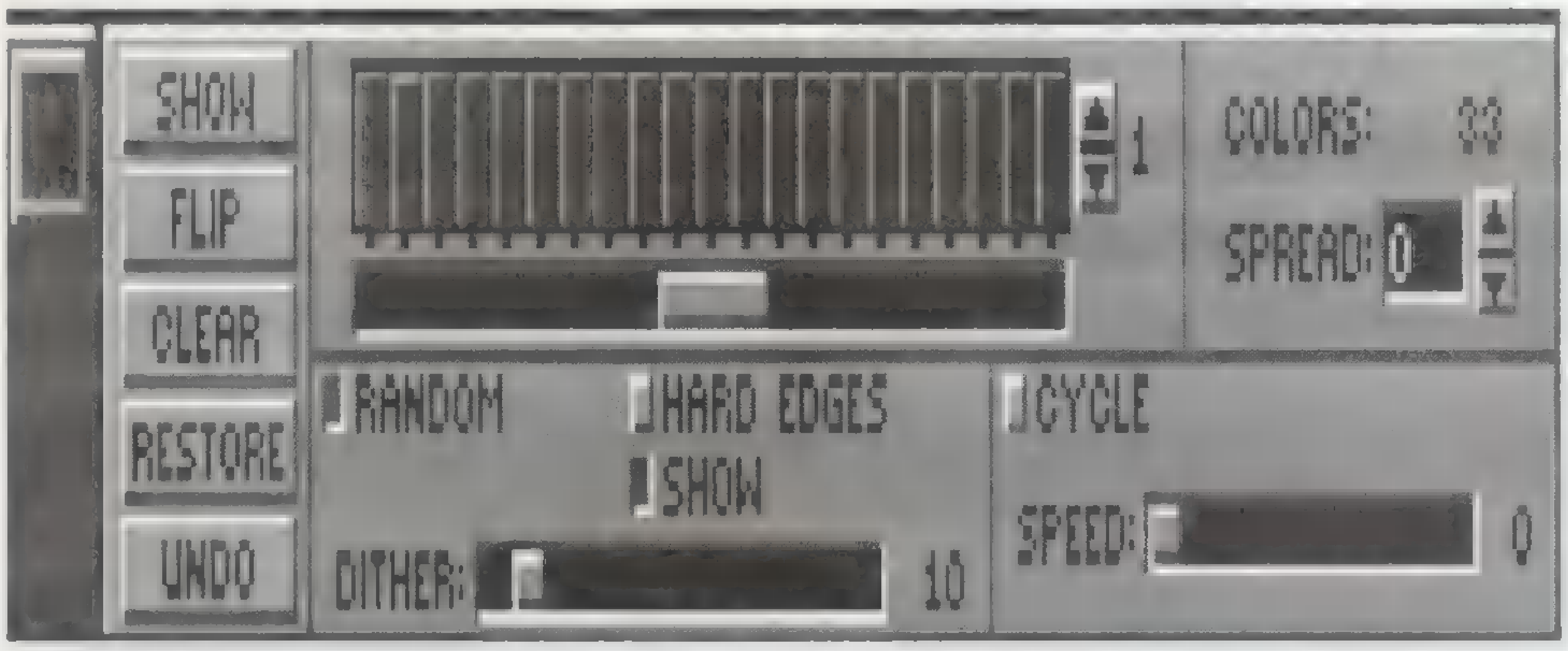


Rysunek 3.

- Load, Save - odczyt i zapis zdefiniowanej palety kolorów,
- Copy - skopiowanie wybranego koloru w dowolne miejsce palety,
- Swap - zamiana miejscami dwóch wskazanych kolorów,
- Spread - pozwala dokonać płynnego przejścia kolorów, pomiędzy aktualnie zaznaczonym, a innym wskazanym przez nas w palecie,
- Default - powrót do standardowej palety,
- Restore - przywrócenie poprzedniej palety,



Rysunek: Cieniowanie.



Rysunek 1.

- Undo - wycofanie się z ostatniej czynności,
- Przyciski R,G,B i C,M,Y - określają rodzaj aktualnie zmienianej składowej kolorów,
- Przyciski HSV,RGB,CMY - określają sposób dobierania składowych kolorów,
- Page - dobiera paletę rysunku zależnie od wybranego koloru,
- Copy, Swap - przekazują kolory spod prawego przycisku myszy w miejsce wskazanej na rysunku barwy bez zmiany palety.

Znając już podstawowe opcje tego menu, przygotujmy sobie odpowiednie, potrzebne przy cieniowaniu, przejścia kolorów. Najpierw zaznaczamy pierwszy z brzegu kolor. Korzystając ze składowych RGB ustawmy (przykładowo): R-255, G-0, B-0. Następnie określamy, przez ile odcieni ma nastąpić przejście kolorów (np. 32) i w odpowiednim miejscu palety, tworzymy nowy kolor: R-100, G-0, B-0. Teraz wracamy na początek, wybieramy opcję "Spread" i wskazujemy kolor końcowy (drugi z ustawionych). Program automatycznie wypełni paletę brakującymi odcieniami.

Przejdźmy do menu "Gradient" (przyciskając prawy klawisz myszy, Rysunek 3). Centralne miejsce zajmuje okno, w którym będziemy określać ilość i rodzaj kolorów biorących udział w cieniowaniu. Aby tego dokonać musimy wskazać w palecie pierwszy ze zdefiniowanych wcześniej kolorów (R-255, G-0,B-0) i przenieść go do wspomnianego wcześniej okna. Naciskając lewy przycisk myszy i przesuwając wskaźnik w prawo, skopiujemy uzyskane wcześniej odcienie. Menu znajdujące się po lewej stronie pozwalają odpowiednio na:

- Show - obejrzenie przejścia kolorów w postaci paska, który pojawi się nad głównym panelem programu,
- Flip - dokonanie lustrzanego odbicia kolorów biorących udział w cieniowaniu,
- Clear - usuwa wszystkie kolory biorące udział w cieniowaniu,
- Restore - przywraca poprzedni stan kolorów,
- Undo - wycofanie się z ostatniej czynności.

Środkową część panelu "Gradient" poświęcono opcji "Dither". Pozwala ona nam na samodzielne operowanie stopniem rozmazania kolorów przy cieniowaniu. Jej możliwości zaprezentowane zostały na Rysunku 4.

Dalej znajduje się opcja "Cycle". Powoduje ona płynne, cykliczne przejścia kolorów w cieniowanych przez nas figurach. Aby w pełni zaobserwować ten efekt należy włączyć opcję "Cycle" i wskazać lewym przyciskiem myszy "Cycle/Gradient" z panelu głównego. Suwakiem "Speed" regulujemy szybkość z jaką następuje powtarzanie cykli kolorów. Prawy, górny róg informuje nas o liczbie kolorów wykorzystanych do cieniowania.

Po odpowiednim ustawieniu wspomnianych opcji możemy przystąpić do tworzenia własnych grafik. Korzystanie wyłącznie z cieniowania nie pozwoli nam na uzyskanie oczekiwanych efektów, dlatego też we wspomnianym wcześniej "Draw mode menu" znalazł się cały szereg dodatkowych opcji graficznych (przykłady ich działania zostały pokazane na Rysunku 5).

Są to odpowiednio:

- Color - pozwala rysować dowolnie wybranym kolorem (opcja jest ustawiona standardowo po uruchomieniu programu),
- Tint - pozwala rysować jedynie tam, gdzie na ekranie znalazł się inny kolor, niż kolor podkładu (tła). Równocześnie następuje zmiana odcienia wcześniej naniesionego koloru (przypomina to nieco mieszanie farb malarskich),
- Colorize - działa podobnie jak -Tint-, przy czym barwy nie są mieszane (wybrany kolor zastępuje istniejący),
- Brighten - rozjaśnianie kolorów na rysunku,
- Darken - ściemnianie kolorów na rysunku,
- Stencil - wszystko, co narysujemy po wybraniu tej opcji będzie za



Rysunek 4.

bezpieczone przed "zamalowaniem". Rozwinięciem tej opcji jest panel "Stencil menu",

- Mix - jak sama nazwa wskazuje, opcja ta pozwala na mieszanie barw na całym rysunku, przy czym obok koloru pędzla, na ostateczny efekt ma również wpływ kolor, od którego rozpoczynany "rozmazywanie",
- Smooth - działa nieco podobnie do opcji "Anty-alias". Powoduje rozmycie konturów oraz zaciera granice między nimi,
- Smear - "rozsmarowuje" kolory na ekranie,
- Range - pozwala rozjaśniać i ściemniać pocieniowaną figurę wykorzystując do tego skrajne kolory użyte w "Gradient menu",
- Cycle - linie będą rysowane z przejściami kolorów ustalonymi w "Gradient menu",
- Random - po wybraniu tej opcji poszczególne punkty lub wycięte brushe wykorzystywane w opcji "Spray", będą rysowane kolejnymi odcieniami z "Gradient menu",
- Dither 1 - rysuje przezroczystą siateczkę, w której stopień prześwitywania ustala się suwakiem "Amount",
- Dither 2 - pozwala narysować nieprzezroczystą, dwukolorową siatkę. Suwakiem "Amount" ustala się stopień nasycenia kolorów (położenie zerowe odpowiada 100% koloru przypisanego prawemu przyciskowi myszy, natomiast 100% na suwaku to 100% udziału w siatce koloru spod lewego klawisza myszy),
- Negative - rysując przy włączonej opcji - Negative - uzyskamy obraz negatywowy poszczególnych kolorów (czarne staje się białym, białe czarnym, itd),
- Halfbrite - Ściemnia o połowę kolory przy wypełnieniach,



Rysunek 5.

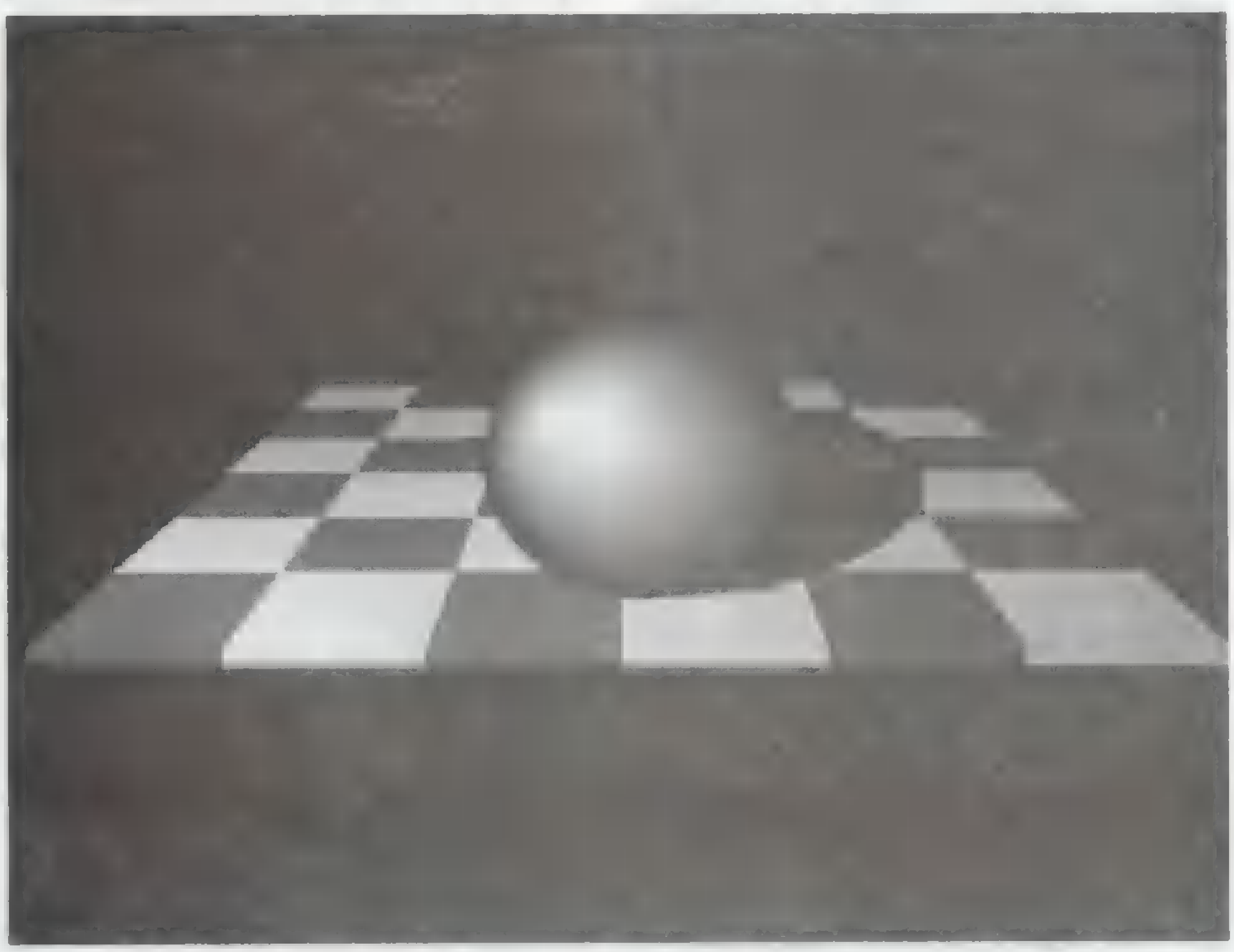
- Not - wykonuje logiczną operację not na kolorach.

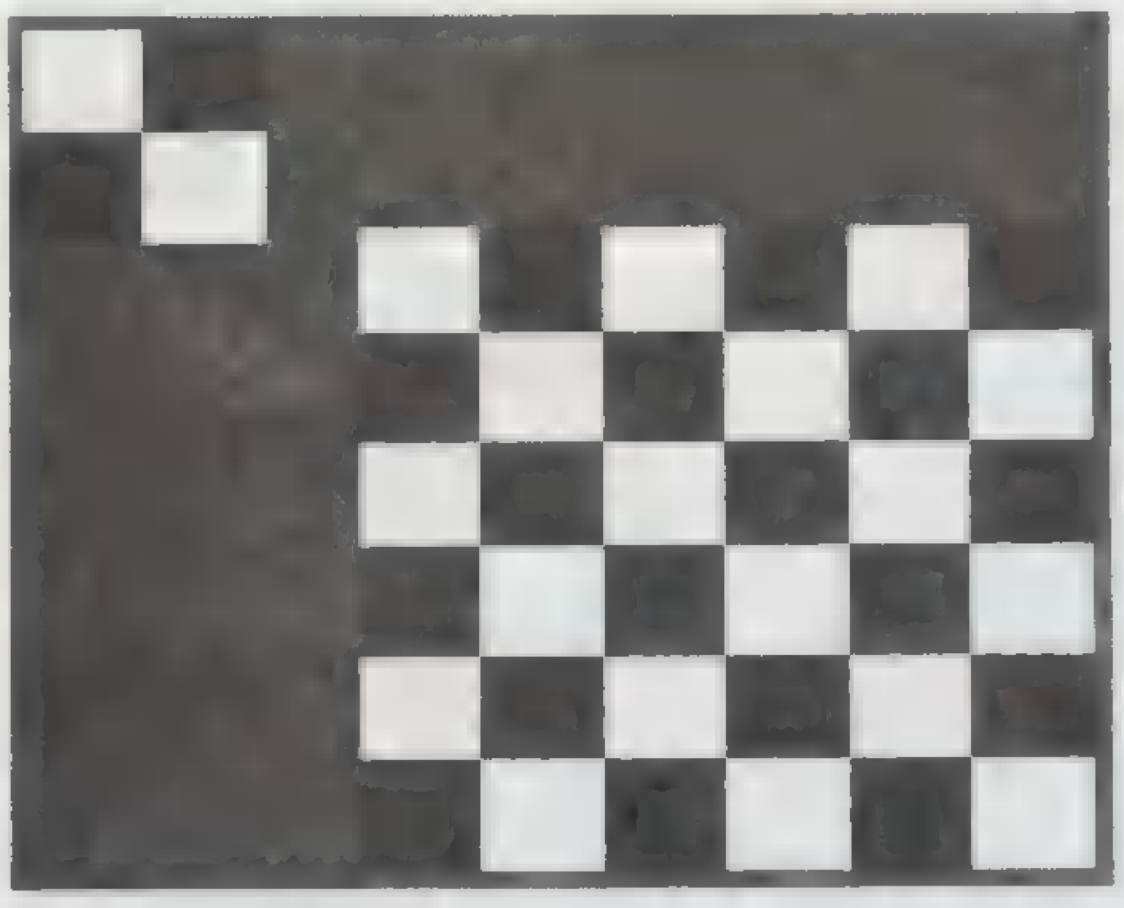
Opcje "Conform" i "Center" pozwalają odpowiednio na: dostosowanie cieniowania do kształtu figury, rozpoczęcie cieniowania figury od jej środka (przy cieniowaniu sposobami: "Highlight", "Spherical" i "Radial").

Pozostał jeszcze do omówienia "Brush fill mode". Zawiera on:

- Stretch - wypełnia rysowaną figurę brushem, dostosowując jego wielkość do wielkości figury,
- Pattern - wypełnia figurę powtarzającym się brushem (o wielkości odpowiadającej oryginalnie wyciętemu brushowi),
- Shape - wypełnia figurę brushem dostosowując jego kształt do kształtu figury,
- Perspective - wypełnia figurę brushem z zachowaniem zdefiniowanych w menu "Tweening" parametrów.

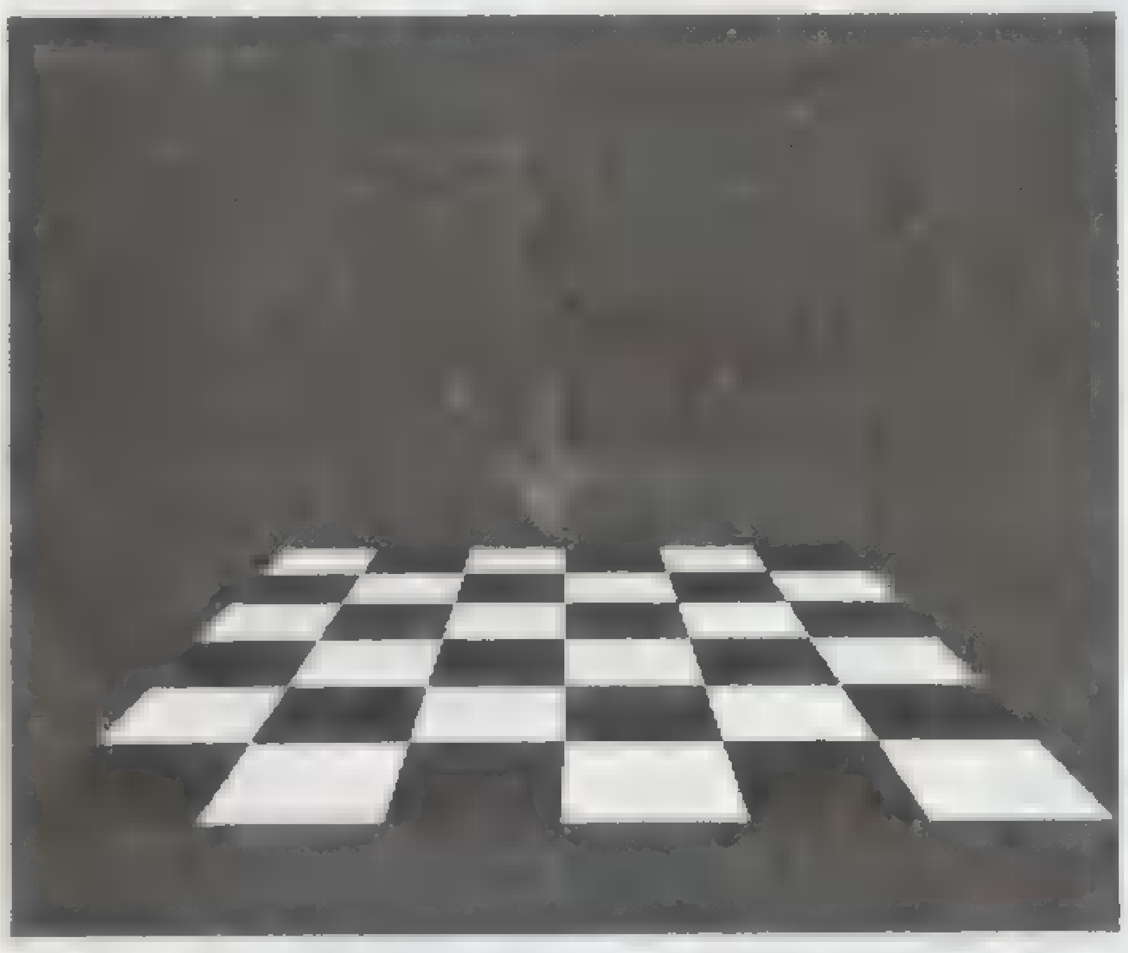
Pora przystąpić do konkretów. Wykonajmy więc prosty, przykładowy rysunek (Rysunek 6). Na początku przygotowujemy podstawowy fragment szachownicy (Rysunek 7a), który następnie wycinamy (brush). Dalej korzystając z opcji "Draw mode"-"Patern" tworzymy całą szachownicę (Rysunek 7b). Tak przygotowany element powtórnie wycinamy (całość) i przechodzimy do "Tweening"-"Adjust". Tutaj za pomocą "Pos"-"X-Y" oraz "Rot"-"X" ustalamy rzut perspektywiczny naszej


Rysunek 6.



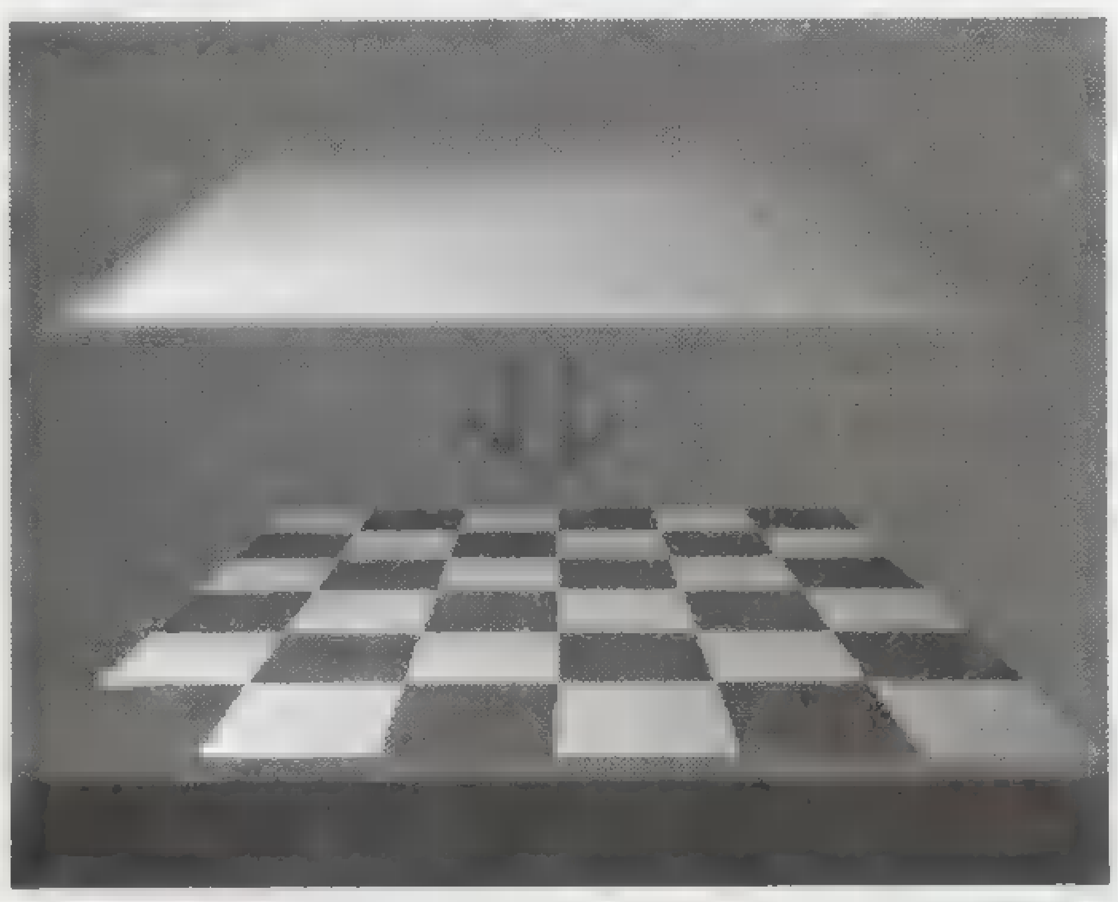
Rysunek 7a-7b.

szachownicy i wybieramy "Use". Po powrocie do panelu "Tweening" ustawiamy "Count" i "Draw" na 1. Wybieramy "Draw" (Rysunek 7c). Teraz kolej na "oświetlenie". Według omówionych wcześniej zasad przygotowujemy przejście kolorów, od jasnoszarego (np. R,G,B-230) do ciemnoszarego

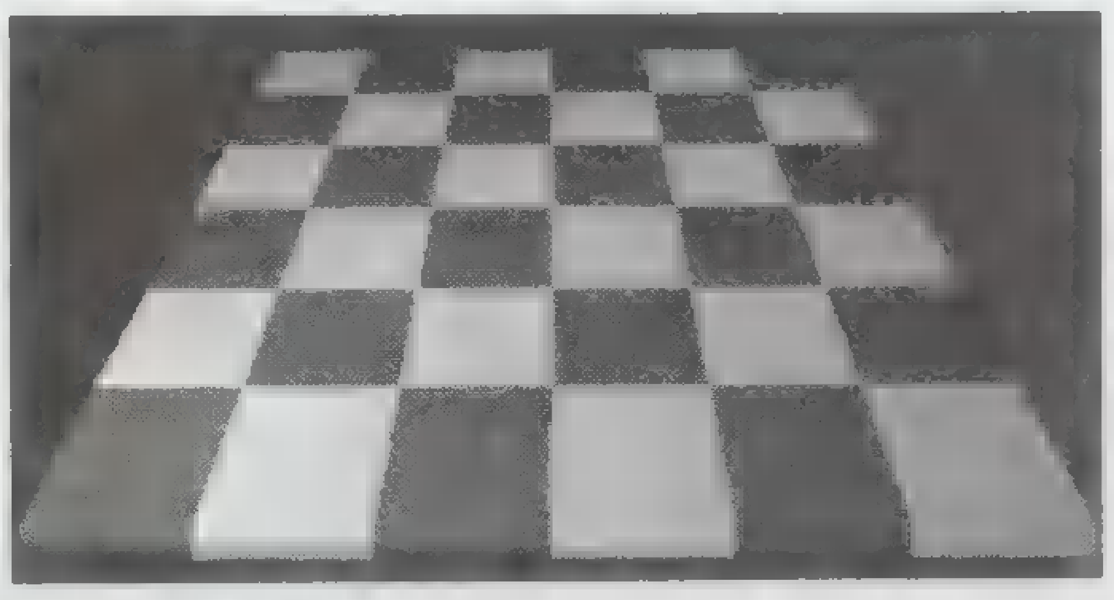


Rysunek 7c.

rego (np. R,G,B-110). W menu "Draw mode" wybieramy "Highlight" i rysujemy figurę odpowiadającą naszej szachownicy (Rysunek 7d). Figurę tę wycinamy jako brusha i po włączeniu opcji "Transparent" nanosimy na szachownicę. Powstaje w ten sposób oświetlona powierzchnia. Do niej, w analogiczny sposób, dorysowujemy przednią krawędź (Rysunek 7e). Teraz kolej na wykonanie cienia kulki. Przechodzimy najpierw do "Ellipse menu" (przycisnąć prawy klawisz myszy na ikonce "Ellipse" panelu głównego) i zaznaczamy "Rotate". Po wybraniu z palety barw dowolnego, ciemnoszarego koloru, rysujemy obróconą elipsę (pozostawiając cały czas włączoną opcję "Transparent" i wyłączając "Draw mode" - Rysunek 7f). Do wykonania pozostała jeszcze oświetlona kulka. Włączamy ponownie "Draw mode" (w menu tym cały czas pozostawała ustawiona opcja "Highlight") i w sposób pokazany na Rysunku 7g "stawiamy" kulkę.

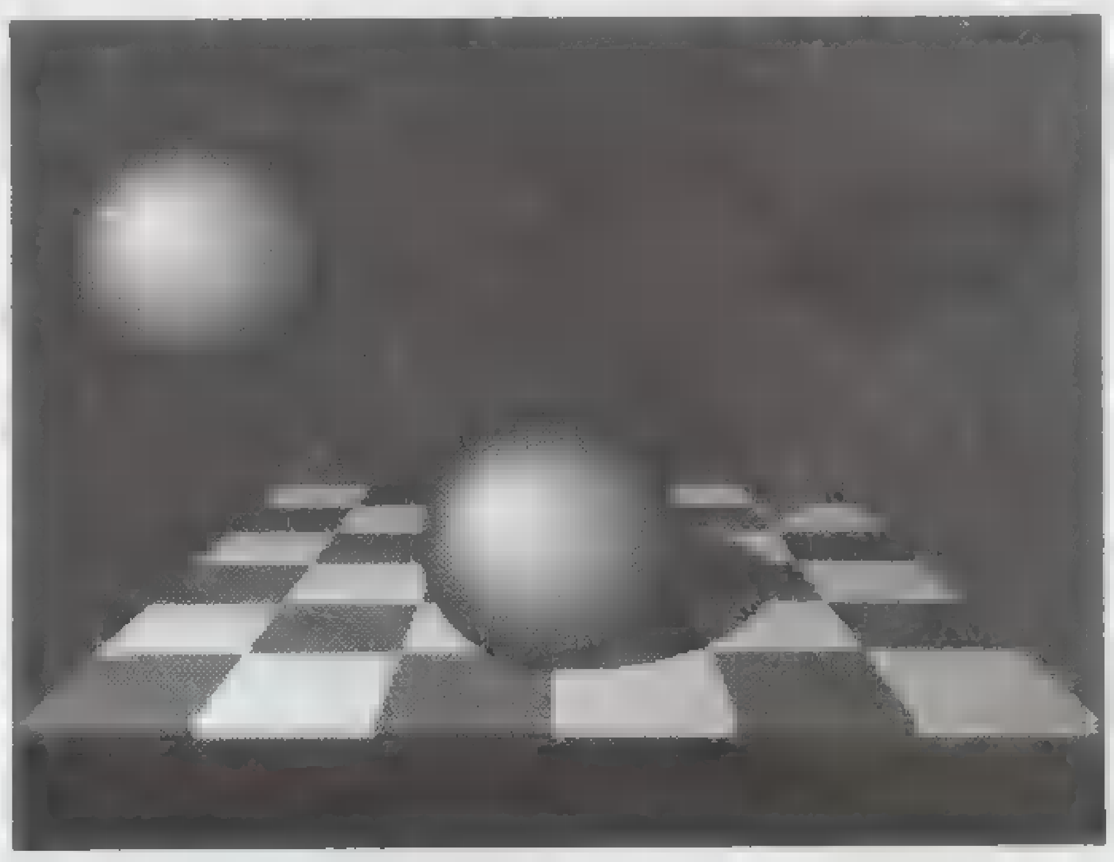


Rysunek 7d.



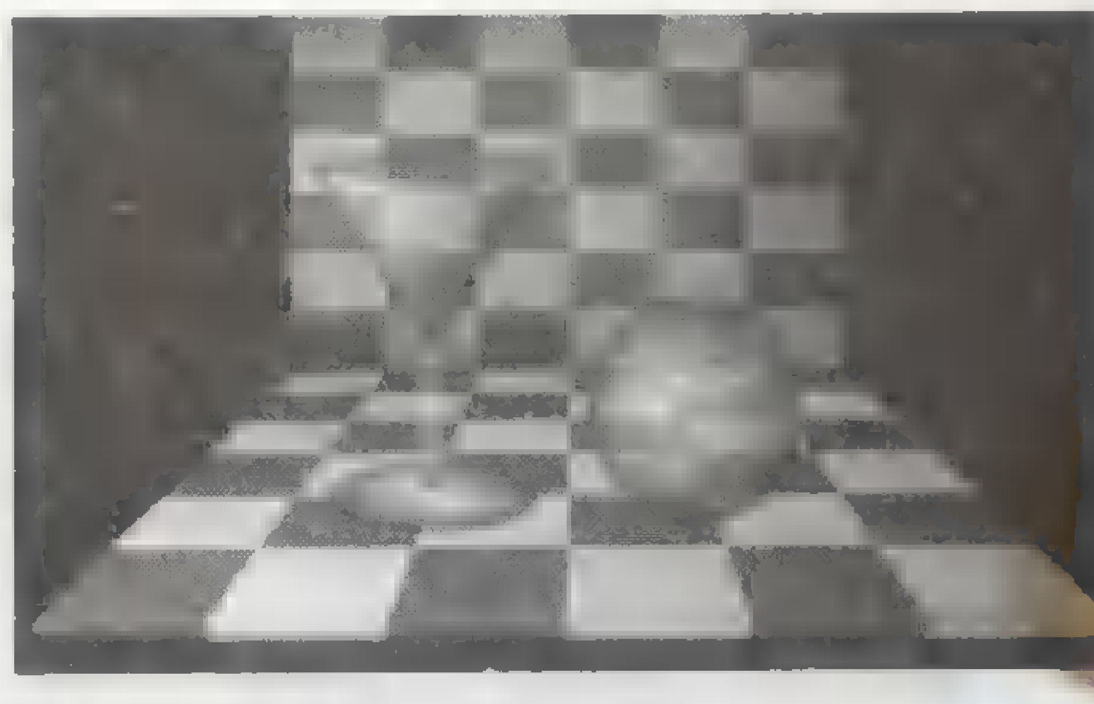
Rysunek 7e.

Kompozycja przedstawiona na Rysunku 8 powstała w bardzo podobny sposób do poprzedniej. Wykorzystano w niej stworzone wcześniej elementy dodając jako tło ścianę, wykonaną w analogiczny sposób jak szachownica. Jest to potrzebne, aby w pełni pokazać przezroczystość znajdujących się na rysunku: kieliszka i kuli. Te dwa nowe elementy zostały wykonane



Rysunek 7f-7g.

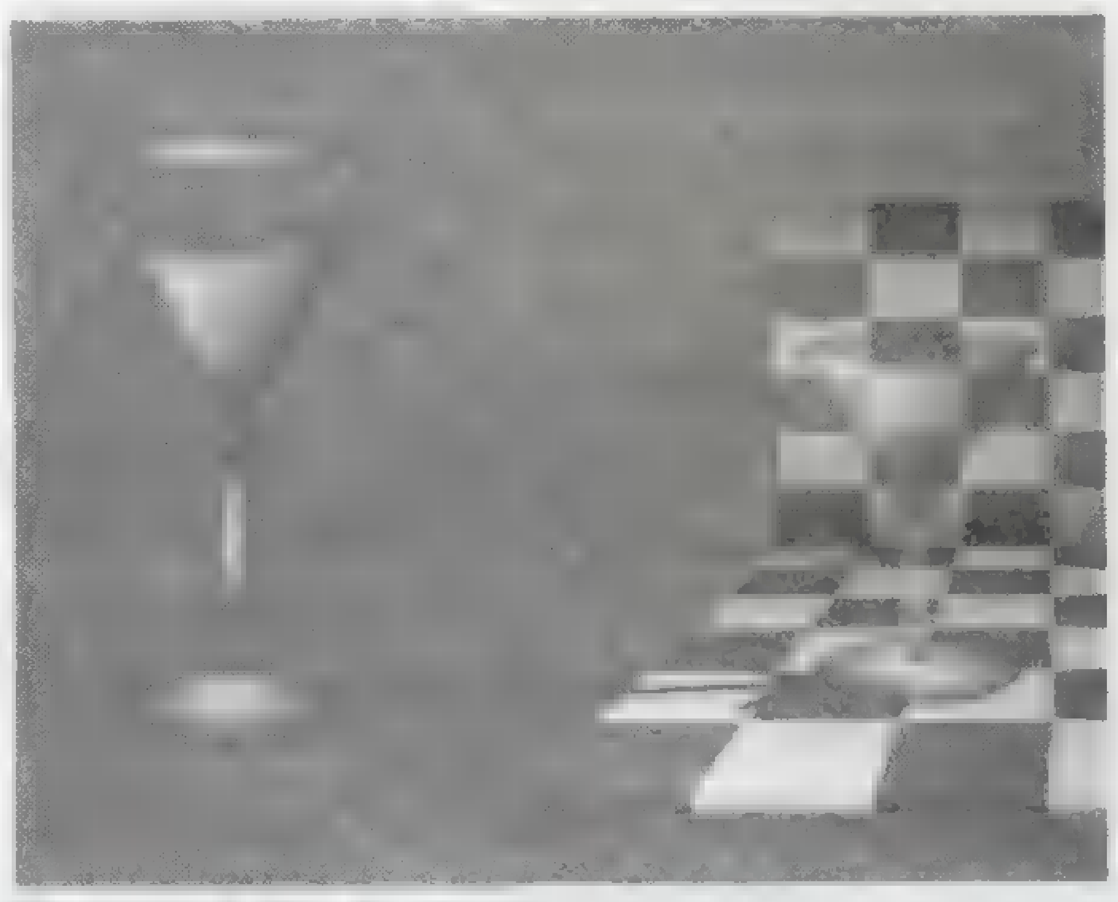
z pomocą znanych nam już opcji "Transparent" i "Highlight". W przypadku kieliszka rozpoczynamy rysowanie od wykonania elipsy (Rysunek 8a), następnie korzystając z "Filled lines" tworzymy nóżkę (Rysunek 8b). Kolejnym elementem jest elipsa (Rysunek 8c) reprezentująca górną krawędź kieliszka. Dla połączenia nóżki z krawędzią wykorzystujemy powtórnie "Filled lines" (Rysunek 8d). Kulkę tworzymy w taki sam sposób jak na Rysunku 7g, co przy włączonej opcji



Rysunek 8.

"Transparent" da efekt prześwitywania.

Kolejna grafika (Rysunek 9) przedstawia możliwości opcji "Smooth" i "Smear". "Przestrzelony balonik" powstał z cieniowanej kulki, w której krawędzie zostały rozmazane opcją "Smear" (wlot i wylot pocisku), a następnie wygładzone za pomocą "Smooth". W podobny sposób został wykonany wylatujący pocisk. Natomiast w znajdującym się poniżej "wybuchu supernowej" wprowadzono podczas rozsmarowywania dodatkowe kolory.

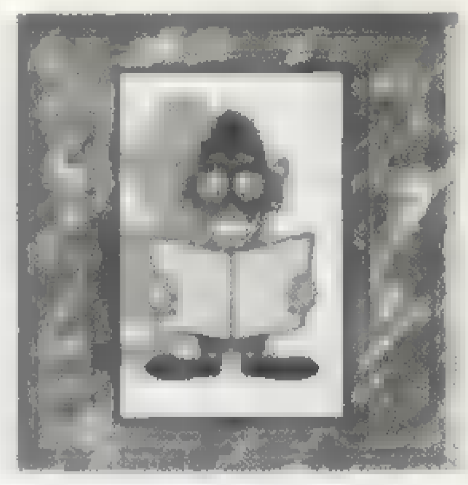


Rysunek 8a, b, c, d.

Na razie to tyle, jeżeli chodzi o drugą część artykułu. Uwzględniając, że udało mi się do tej pory zasygnalizować jedynie ok. 70% opcji programu, nie trudno sobie wyobrazić, jak potężnym narzędziem jest "Brilliance". ❑



Rysunek 9.



AMIGA Dos 2.xi3.x

nowe odkrycie

[illegible]

Polska instrukcja - to rzecz, której najbardziej pożąda początkujący właściciel Amigi. Wydawać by się mogło, że tę zachciankę powinien spełnić producent komputera. Jednak ze względu na bezwładność kolusa (czyt. Commodora) pojawiają się publikacje mające uzupełnić oryginalną instrukcję komputera.

Ze względu na brak polskich instrukcji obsługi w znakomitej większości komputerów sprzedawanych w Polsce użytkownicy Amigi, którzy nie potrafią posługiwać się "obcą mową" są całkowicie odcięci od informacji w jaki sposób to swoje cudo obsługiwać. Znajomość angielskiego lub niemieckiego też nie zawsze jest przepustką otwierającą bramę wiedzy, ponieważ do Polski trafiają Amigi wyposażone w instrukcje włoskie, hiszpańskie, itp. (szczytem wszystkiego jest zakup komputera przeznaczonego dla Holandii - próba przeczytania dołączonej do komputera instrukcji dla większości Polaków może zakończyć się zwichnięciem szczęki).

Oczywiście "dziurę" rynkową zauważyli już dawno polscy autorzy i wydawcy, co zaowocowało wydaniem kilku mniej lub bardziej udanych instrukcji posługiwania się Amigą. Jedną z najświeższych prób podjęcia tego tematu jest seria pięciu (przynajmniej tyle do nas dotarło) książek wydawnictwa Andromeda opracowanych przez pana Andrzeja Pawłowskiego. Są to następujące tytuły: Workbench 1.3, Workbench 2.0, AmigaDOS 2.0 oraz co zapewne zainteresuje posiadaczy Amigi 1200 i 4000: Workbench 3.0 i AmigaDOS 3.0.

Już same tytuły mówią o zawartości tych pozycji wydawniczych, zgodnie z którymi opisują one odpowiednie wersje systemów operacyjnych stosowanych w różnych modelach Amigi. Opisywane książki swoim układem przypominają oryginalną instrukcję firmy Commodore.

Podręczniki Workbencha rozpoczynają się krótkim wstępem opisującym sposób połączenia komputera ze światem zewnętrznym (monitory, klawiatura, myszka, zasilacz, itp.). Koniec pierwszego rozdziału poświęcony jest podstawowym zasadom posługiwania się komputerem. Następnie przechodzimy do drugiej części podręcznika, w której autor tłumaczy podstawowe pojęcia związane z komunikacją człowiek z komputerem (wskaźnik myszki, klawisze specjalne i funkcyjne, ikony, okna, itp.). Dalsze części książki poświęcone są kolejno: podstawowym operacjom związanym z pracą "na Workbenchu" (poczynając od przesuwania okienek, a kończąc na kopiowaniu dyskietek i obsłudze RAM-dysku), funkcjom wykonywanym z zastosowaniem menu Workbencha, korzystaniu z klawiatury. Następnie autor opisuje sposoby konfigurowania Workbencha do własnych potrzeb oraz korzystania z niektórych programików dołączonych przez firmę Commodore do systemu. Sporo miejsca poświęcono ikonom oraz ukrytym informacjom w nich. Nie zabrakło opisu podstawowych informacji o AmigaDOS (przydadzą się szczególnie opisy najczęściej występujących w systemie błędów). Książkę kończy opis niektórych urządzeń peryferyjnych, oraz krótki samouczek posługiwania się dyskiem twardym.

Z nieco mniejszej ilości części składają się podręczniki AmigaDOSu. Oprócz najbardziej rozbudowanego rozdziału zawierającego opis składni i działania komend DOSu możemy poznać podstawowe edytory służące do opracowywania własnych programów (EDIT, ED, MEMACS). Te dwa rozdziały stanowią najbardziej rozbudowaną część podręczników. Oczywiście nie miałyby one większej wartości dla początkujących, gdyby ich treść nie została poprzedzona odpowiednim wstępem. Taki wstęp istnieje i zawiera podstawowe informacje o organizacji samego systemu, wykorzystywanych w nim urządzeniach i sposobie odwoływania się do nich, nazwach plików i katalogów, zawartości niektórych katalogów wykorzystywanych przez komputer. Książki kończy opis błędów (również tych, które generuje GURU) oraz dodatki zawierające kody ASCII, podstawowe komendy opisywanych edytorów oraz spis komend DOSu.

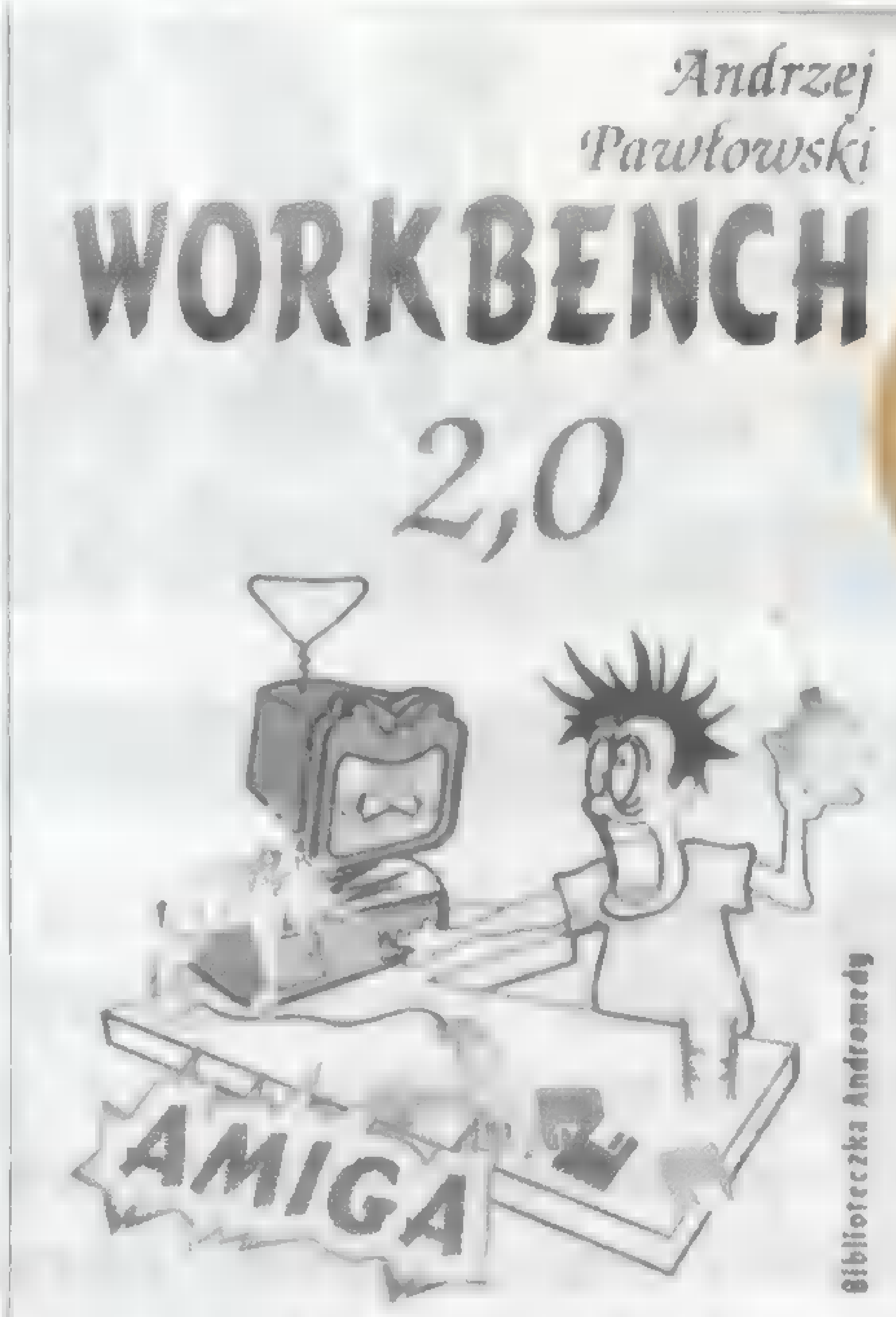


Zawartość poszczególnych rozdziałów nie budzi większych zastrzeżeń (o tych mniejszych jeszcze porozmawiamy). "Pierwsze czytanie" redakcyjne nie zaowocowało znalezieniem błędów, które mogłyby doprowadzić początkującego posiadacza Amigi do rozstroju nerwowego. Warto również zauważyć, że książki jak na warunki polskie są starannie wydane (nie należy spodziewać się wodotrysków, ale w końcu przyjemnie jest popatrzeć na książkę, która nie sprawia wrażenia odbitki kserograficznej).

To co napisałem o tej serii wydawnictwa Andromeda wystarczy, aby zorientować się mniej więcej jakie informacje możemy w niej znaleźć. Teraz przejdziemy do nieco mniej przyjemnej części mojego zadania, a mianowicie wytykania błędów.

Pierwszy z nich jest wymieniany zwykle w różnych publikacjach dotyczących książek dla początkujących użytkowników (na dowolny temat). Dotyczy to szczególnego zadania, które ma autor. Książka dla osoby uczącej się powinna stopniowo wprowadzać nowe elementy ze szczególnym naciskiem na opanowanie podstaw. Zdaję sobie sprawę, że dla osoby, która wszystkie problemy związane z opanowaniem komputera ma już za sobą (a takie osoby są najczęściej autorami książek) jest to bardzo trudne. Panu Andrzejowi Pawłowskiemu nie udało się niestety do końca dokonać powrotu do podstaw swojej wiedzy. Jest to widoczne szczególnie przy opisie Workbencha, moim zdaniem część poświęcona początkom współpracy z komputerem jest zbyt uboga. Brakuje w niej przede wszystkim przykładów pozwalających początkującemu użytkownikowi przełamać lęk przed komputerem. Oprócz nich warto byłoby się zastanowić nad zobrazowaniem treści opisów większą ilością grafiki. Ogólnie rzecz biorąc opisy Workbencha zbyt wiernie naśladują oryginalną instrukcję firmy Commodore, która moim zdaniem nie stanowi najlepszego wzoru. Nieco lepiej ten problem ma się w przypadku opisów AmigaDOSu. Rażąca w przypadku poprzednich podręczników hasłowość jest tutaj pomocna. Umożliwia szybkie znalezienia definicji potrzebnej w danym momencie komendy. Jednakże, tak jak w przypadku książek o Workbenchu, tak i w przypadku AmigaDOSu muszę zarzucić autorowi zbyt skromny udział rozdziałów poświęconych początkującym oraz brak przykładów.









Na koniec muszę niestety obdarować wszystkich tzw. podsumowaniem ("rednacz" nie przyjmuje artykułów bez zakończenia). Książki pana Pawłowskiego zawierają wiedzę przydatną z pewnością wszystkim użytkownikom Amigi, którzy do tej pory nie opanowali możliwości swojego komputera. Początkujący będą jednak musieli liczyć się z koniecznością częstego wertowania kartek w poszukiwaniu potrzebnej informacji oraz wykazać sporo cierpliwości i sprytu przy przekształcaniu teorii zawartej w podręcznikach na umiejętności praktyczne.

Książki Andrzeja Pawłowskiego z serii Biblioteczka Andromedy: Workbench 1.3, Workbench 2.0, Workbench 3.0, AmigaDOS v.2.0, AmigaDOS v.3.0, uzyskaliśmy dzięki uprzejmości:

Wydawnictwa Andromeda
00-976 Warszawa 13
skr. poczt. 12

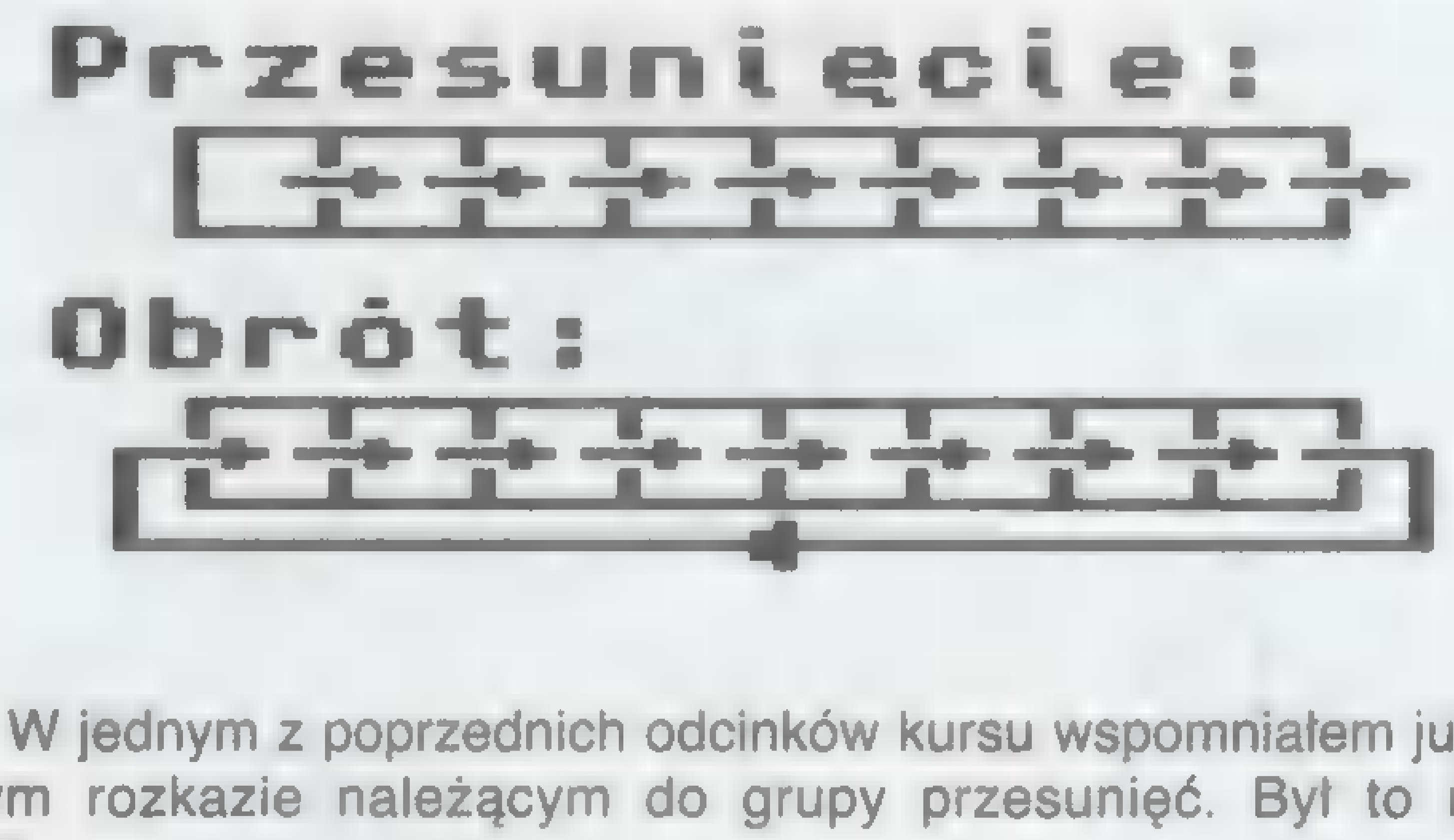
ceny:
"Workbench 1.3", "Workbench 2.0", "Workbench 3.0" - 55 tys. zł, każda, "Dos 2.0", Dos 3.0" -60 tys. zł, każda,

KURS ASEMBLERA

Adam Gregorowicz

W dzisiejszym odcinku kursu będziemy przesuwać i obracać. To na pozór bezsensowne stwiedzenie oznacza, że poznacie osiem rozkazów należących do grupy instrukcji realizujących przesunięcia i obroty.

Oczywiście rozkazy przesunięć i obrotów nie powodują przesuwania lub obracania procesora albo komputera na stole - dokonują po prostu przesuwania bitów wewnątrz bajtu, co najłatwiej zrozumieć na małym przykładzie. Otóż przykładowe przesunięcie bitów w bajcie w prawo o jeden bit oznacza, że zawatość bitu 1 zostanie skopiowana do bitu 0, bitu 2 - do bitu 1, bitu 3 - do 2, itd. Co zaś dzieje się z zawartością bitu 0? To zależy. Jeśli dokonujemy tylko przesunięcia, to zostaje ona stracona (nie jest to dokładnie prawdą, ale o tym później), gdy jednak dokonujemy obrotu, zawartość tego bitu trafia do bitu 7 (w przypadku obrotu bajtu). Różnice między przesunięciem, a obrotem najłatwiej jest przedstawić graficznie (to znaczy za pomocą obrazka).



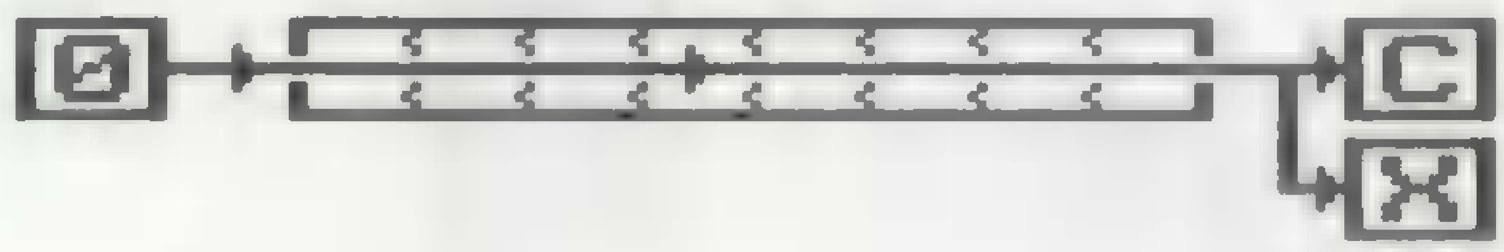
W jednym z poprzednich odcinków kursu wspomniałem już o jednym rozkazie należącym do grupy przesunięć. Był to rozkaz LSR.

LSR - Logical shift right (Logiczne przesunięcie w prawo.)

Zapis: LSR #n,Dn , LSR przeznaczenie lub LSR Dn,Dm

Rozkaz LSR dokonuje przesunięcia bitów operandu przeznaczenia w prawo. Możliwe jest przesunięcie jednym rozkazem o 1 do 8 bitów, jednak tylko wtedy, gdy operandem przeznaczenia jest rejestr danych (LSR #n,Dn). Zawartość rejestru danych można również przesunąć o liczbę bitów zawartą w innym rejestrze danych - LSR Dn,Dm - gdzie Dn jest rejestrem zawierającym informację: o ile bitów przesunąć zawartość rejestru Dm. Oprócz tego możliwe jest przesunięcie zawartości pamięci adresowanej w sposób bezpośredni lub za pomocą rejestru adresowego, lecz tylko o jeden bit.

Rozmiarem operacji może być bajt, słowo lub długie słowo, gdy przesuwamy zawartość rejestru danych. Natomiast w przypadku przesuwania zawartości pamięci rozmiarem operacji może być tylko słowo.



Działanie rozkazu LSR.

Jak widać z rysunku przedstawiającego działanie intrukcji LSR, zawartość najmłodszego bitu przesuwanego operandu jest kopiowana do znaczników C oraz X. Natomiast najstarszy bit jest zerowany (jest tam jakby wsuwane zero).

Znaczniki:
- **X** - Ustawiany zgodnie z ostatnim wysuniętym bitem.
- **N** - Ustawiany, gdy najstarszy bit wyniku jest równy jeden. W przeciwnym razie zerowany.
- **Z** - Ustawiany, gdy wynikiem operacji jest zero. W przeciwnym wypadku zerowany.
- **V** - Zawsze zerowany.
- **C** - Ustawiany zgodnie z ostatnim wysuniętym bitem (dokładnie tak samo jak znacznik X).

*** Dla rozkazu LSR przeznaczenie umożliwiającego przesunięcie zawartości pamięci o jeden bit dozwolone są następujące tryby adresowania:

```
LSR (An)
LSR (An)+
LSR -(An)
LSR p(An)
LSR p(An,Rm.W)
LSR p(An,Rm.L)
LSR nn.W
LSR nn.L
```

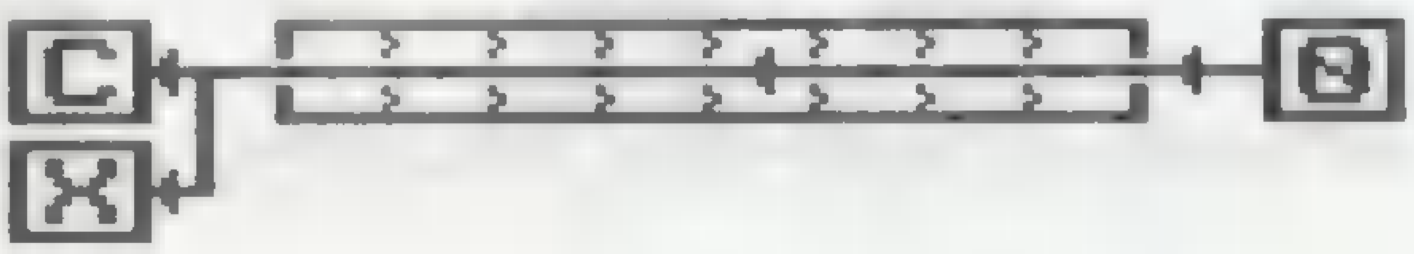
Przykład: LSR.B #1,D0 powoduje przesunięcie zawartości rejestru danych D0 o jeden bit w prawo. Gdyby przykładowo w rejestrze D0 była wartość %10011010, to po wykonaniu instrukcji LSR #1,D0 znalazłaby się tam wartość %01001101, a znaczniki C i X zostałyby wyzerowane.

Przykład: LSR (A5)+ powoduje przesunięcie o jeden bit w prawo słowa w pamięci pod adresem zawartym w rejestrze adresowym A5. Po dokonaniu przesunięcia zawartość rejestru A5 zostaje zwiększona o 2 (tryb adresowania z post inkrementacją). Jak widać nie został tu podany rozmiar operacji, gdyż jest ona możliwa jedynie na słowie (równie dobrze można było napisać LSR.W (A5)+).

Oczywiście skoro istnieje rozkaz przesuwający bity w prawo, to powinien istnieć rozkaz robiący to samo, ale w lewo. Rozkazem takim jest LSL.

LSL - Logical shif left (Logiczne przesunięcie w lewo.)

Zapis: LSL #n,Dn , LSL przeznaczenie lub LSL Dn,Dm
Rozkaz LSL wykonuje dokładnie tę samą operację co LSR, lecz nie w prawo, a w lewo. Przy stosowaniu instrukcji LSL obowiązują te same ograniczenia, co w przypadku LSR. Zachowanie znaczników jest także identyczne jak przy LSR.



Działanie rozkazu LSL.

Przy stosowaniu rozkazu LSL dopuszczalne są te same tryby adresowania, co przy LSR, czyli:

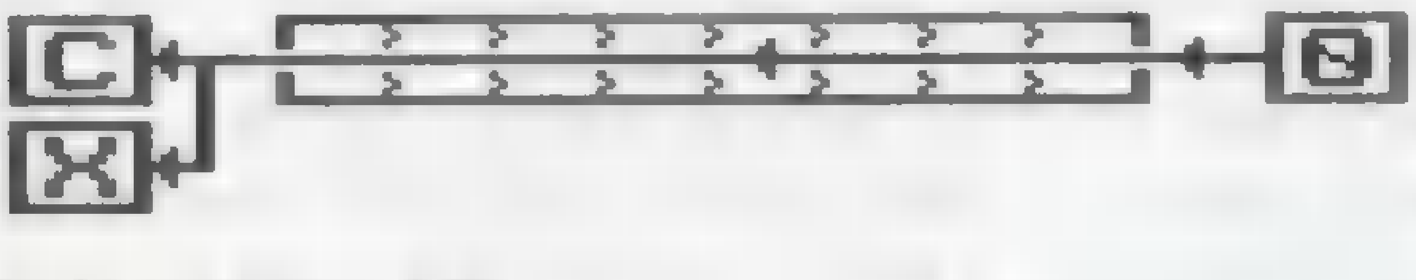
LSL (An)
LSL (An)+
LSL -(An)
LSL p(An)
LSL p(An,Rm.W)
LSL p(An,Rm.L)
LSL nn.W
LSL nn.L

Innym rozkazem z grupy przesunięć jest ASL.

ASL - Arithmetic shift left (Arytmetyczne przesunięcie w lewo.)

Zapis: ASL #n,Dn , ASL przeznaczenie lub ASL Dn,Dm
Instrukcja ASL dokonuje przesunięcia bitów operandu przeznaczenia w lewo. Możliwe jest przesunięcie jednym rozkazem o 1 do 8 bitów. Następuje to jednak tylko wtedy, gdy operandem przeznaczenia jest rejestr danych (ASL #n,Dn). Zawartość rejestru danych można również przesunąć o liczbę bitów zawartą w innym rejestrze danych - ASL Dn,Dm - gdzie Dn jest rejestrem zawierającym informację: o ile bitów przesunąć zawartość rejestru Dm. Oprócz tego możliwe jest przesunięcie zawartości pamięci, lecz tylko o jeden bit.

Rozmiarem operacji może być bajt, słowo lub długie słowo, gdy przesuwamy zawartość rejestru danych. Natomiast w przypadku przesuwania zawartości pamięci rozmiarem operacji może być jedynie słowo.



Działanie rozkazu ASL.

Zaraz, zaraz! Czy rozkaz ASL nie ma identycznego działania jak rozkaz LSL? Tak, ASL jest w pełni zgodny z instrukcją LSL, posiada również takie same jak LSL dozwolone tryby adresowania. W takim razie, po co zdublowano ten rozkaz. Stało się tak dlatego, że arytmetyczne (ASL) i logiczne (LSL) przesunięcie bitów w lewo wygląda dokładnie tak samo. Dopiero przy przesuwaniu w prawo widoczne są różnice. W takim razie, co to jest przesunięcie arytmetyczne? Otóż rozkazy przesunięć można stosować jako uproszczone (i dużo szybsze) rozkazy mnożenia lub dzielenia przez liczby będące potęgami dwójki. Każde przesunięcie o jeden bit w lewo to mnożenie o 2, a przesunięcie w prawo - dzielenie przez 2. (Sprawdźcie, że to prawda.)

Rozkazy przesunięć arytmetycznych służą głównie do dzielenia i mnożenia liczb ze znakiem zapisanym w kodzie uzupełnień do dwóch (opis tego kodu znajduje się w pierwszym odcinku tego kursu), gdzie znak liczby zapisywany jest w najstarszym bicie, tak więc bit ten należy chronić przed zmianami. Co prawda rozkaz ASL nie czyni tego zbyt dobrze, ale za to jego brat - ASR - przesuwający bity arytmetycznie w prawo jest mistrzem w ochronie najstarszego bitu przed zmianami.

ASR - Arithmetic shift right (Arytmetyczne przesunięcie w prawo.)

Zapis: ASR #n,Dn , ASR przeznaczenie lub ASR Dn,Dm
Instrukcja ASR dokonuje arytmetycznego przesunięcia bitów operandu przeznaczenia w prawo. Tak jak w przypadku LSR, LSL, ASL istnieją trzy odmiany tej instrukcji:
- możliwe jest przesunięcie zawartości rejestru danych jednym rozkazem o 1 do 8 bitów (ASR #n,Dm),
- przesunięcie zawartości rejestru danych o liczbę bitów zawartą w innym rejestrze danych - ASR Dn,Dm - gdzie Dn jest rejestrem zawierającym informację: o ile bitów przesunać zawartość rejestru Dm,
- przesunięcie zawartości pamięci o jeden bit.

Rozmiarem operacji może być bajt, słowo lub długie słowo, gdy przesuwamy zawartość rejestru danych, w razie przesuwania zawartości pamięci rozmiarem operacji może być tylko słowo.



Działanie rozkazu ASR.

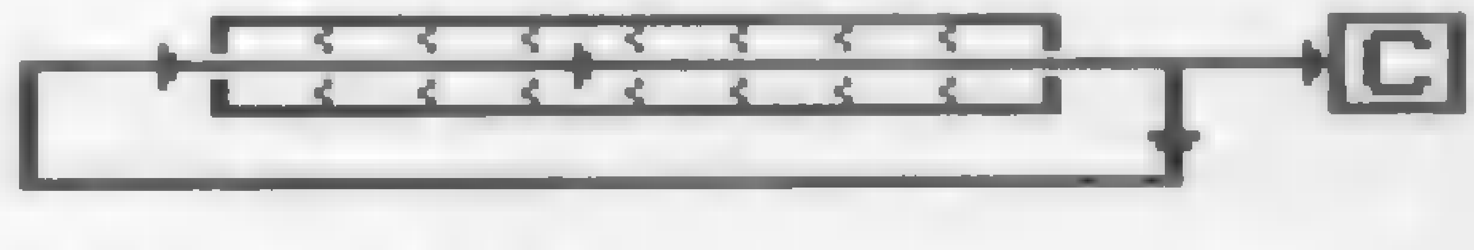
Jak widać z rysunku, przy wykonywaniu instrukcji ASR najstarszy bit jest kopiowany do rejestru leżącego na prawo od niego (o niższym numerze), ale on sam jest nienaruszany, co powoduje zachowanie znaku liczby, którą poddaliśmy przesunięciu.

Zarówno zachowanie znaczników jaki i dozwolone tryby adresowania są identyczne jak w przypadku wszystkich opisanych dotychczas rozkazów z grupy przesunięć.

To już wszystkie instrukcje realizujące przesunięcia, pozostały Nam jednak jeszcze rozkazy obrotów. Jednym z nich jest rozkaz ROR.

ROR - Rotate right (Obrót w prawo.)

Zapis: ROR #n,Dn , ROR przeznaczenie lub ROR Dn,Dm
Rozkaz ROR wykonuje obrót bitów operandu przeznaczenia w prawo. Umożliwia on obrót o 1 do 8 bitów, gdy operandem przeznaczenia jest rejestr danych (ROR #n,Dn). Zawartość rejestru danych można również obrócić o liczbę bitów zawartą w innym rejestrze danych - ROR Dn,Dm - gdzie Dn jest rejestrem zawierającym informację: o ile bitów obrócić zawartość rejestru Dm. Prócz tego możliwy jest obrót zawartości pamięci o jeden bit.



Działanie rozkazu ROR.

Rozmiarem operacji ROR może być bajt, słowo lub długie słowo, gdy obracana jest zawartość rejestru danych. Natomiast w przypadku obrotu bitów w pamięci rozmiarem może być tylko słowo.

Jak wynika z rysunku rozkaz ROR obracając bity dokonuje zwykłego przesunięcia w prawo. Jedynymi różnicami między ROR i LSR jest to, że instrukcja ROR wpisuje najmłodszy wysunięty bit do najstarszego (LSR wpisywało tam 0), powodując jakby "zapętlenie" bitów. ROR różni się od rozkazu przesuniecia również tym, że wartość wysuniętego z prawej bitu jest wpisywana tylko do znacznika C, a nie do C i X, jak to miało miejsce przy wykonaniu instrukcji LSR.

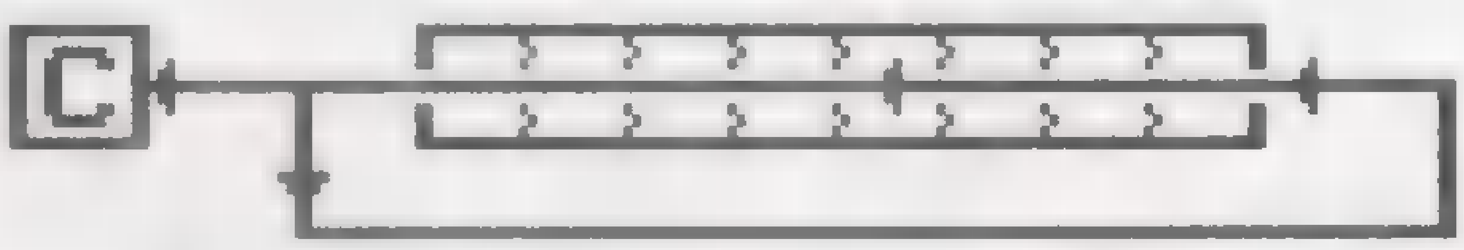
Znaczniki:
X - Nie zmieniany.

N - Ustawiany, gdy najstarszy bit wyniku jest równy jeden. W przeciwnym razie zerowany.
Z - Ustawiany, gdy wynikiem operacji jest zero. W przeciwnym wypadku zerowany.
V - Zawsze zerowany.
C - Ustawiany zgodnie z ostatnim wysuniętym bitem.

Dla rozkazu ROR dozwolone są następujące tryby adresowania:

ROR (An)
ROR (An)+
ROR -(An)
ROR p(An)
ROR p(An,Rm.W)
ROR p(An,Rm.L)
ROR nn.W
ROR nn.L

Przykład: ROR.B #2,D1 powoduje przesunięcie ośmiu bitów (intrukcja ma rozmiar bajtu) zawartości rejestru danych D1 o dwa bity w prawo. Gdyby przykładowo rejestr D1 zawierał wartość %10111001, to po wykonaniu instrukcji ROR.B #2,D1 znalazłaby się tam wartość %01101110, a znacznik C zostałby wyzerowany.



Działanie rozkazu ROL.

Odpowiednikiem instrukcji ROR, który obraca bity nie w prawo, lecz w lewo jest ROL.

ROL - Rotate left (Obrót w lewo.)

Zapis: ROL #n,Dn , ROL przeznaczenie lub ROL Dn,Dm
Rozkaz ROL realizuje tę samą operację co ROR, lecz w przeciwnym kierunku, czyli w lewo. Przy stosowaniu instrukcji ROL obowiązują te same ograniczenia, co w przypadku ROR. Zachowanie znaczników jest także identyczne jak przy ROR.

Używając rozkazu ROL możemy stosować następujące tryby adresowania:

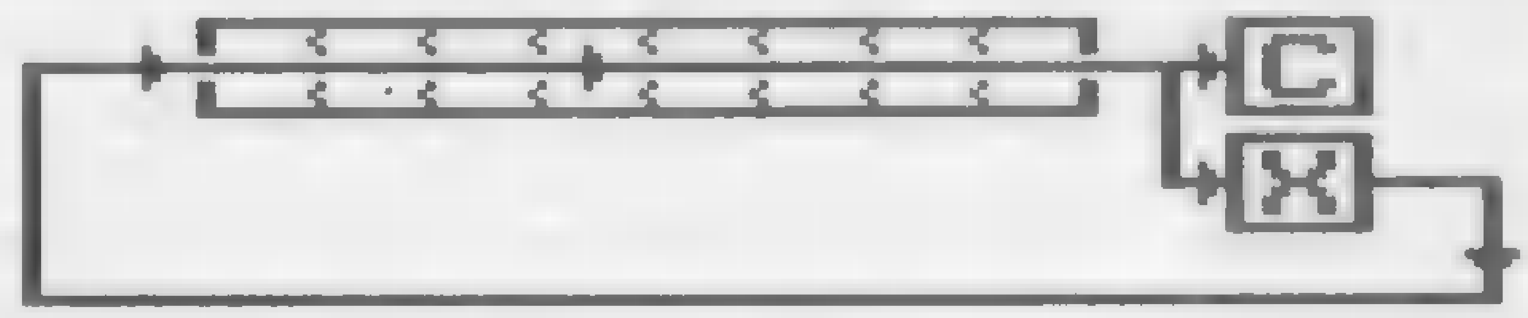
ROL (An)
ROL (An)+
ROL -(An)
ROL p(An)
ROL p(An,Rm.W)
ROL p(An,Rm.L)
ROL nn.W
ROL nn.L

Przykład: ROL 10(A3) powoduje obrót o jeden bit w lewo słowa w pamięci pod adresem, który jest równy zawartości rejestru adresowego A3 powiększonej o 10. Jak przy wszystkich rozkazach przesunięć i obrotów, instrukcja ROL dokonująca obrotu zawartości pamięci może mieć tylko i wyłącznie rozmiar słowa, nie jest więc konieczny zapis ROL.W.

Ostatnimi rozkazami z grupy przesunięć i obrotów, jakimi dyspunuje mikroprocesor Motorola 68000, są ROXR i ROXL.

ROXR - Rotate right with extend (Obrót w prawo z rozszerzeniem.)

Zapis: ROXR #n,Dn , ROXR przeznaczenie lub ROXR Dn,Dm
Rozkaz ROXR wykonuje obrót bitów operandu przeznaczenia w prawo. Tak samo jak i przy innych rozkazach obrotów istnieją trzy warianty instrukcji ROXR:
- obrót o 1 do 8 bitów, gdy operandem przeznaczenia jest rejestr danych (ROXR #n,Dn),
- obrót zawartości rejestru danych o liczbę bitów zawartą w innym rejestrze danych (ROXR Dn,Dm),
- obrót słowa w pamięci o jeden bit.
Rozmiarem operacji ROXR może być bajt, słowo lub długie słowo przy obracaniu zawartości rejestru danych. Słowo w wypadku, gdy obracana jest zawartość pamięci.



Działanie rozkazu ROXR.

Z rysunku opisującego instrukcję ROXR widać, że znacznik C zachowuje się tak samo jak przy innych rozkazach z grupy przesunięć i obrotów, natomiast znacznik X działa jako bit dodatkowy. Na przykład podczas obrotu w rozmiarze bajtu najmłodszy, wysunięty bit nie trafi bezpośrednio do najstarszego bitu obracanego bajtu, lecz zostanie zapamiętany w znaczniku X, którego zawartość zostanie bezpośrednio przed tym wpisana właśnie do najstarszego bitu obracanego bajtu. Daje to jakby "wydłużenie" obracanej wartości o jeden bit.

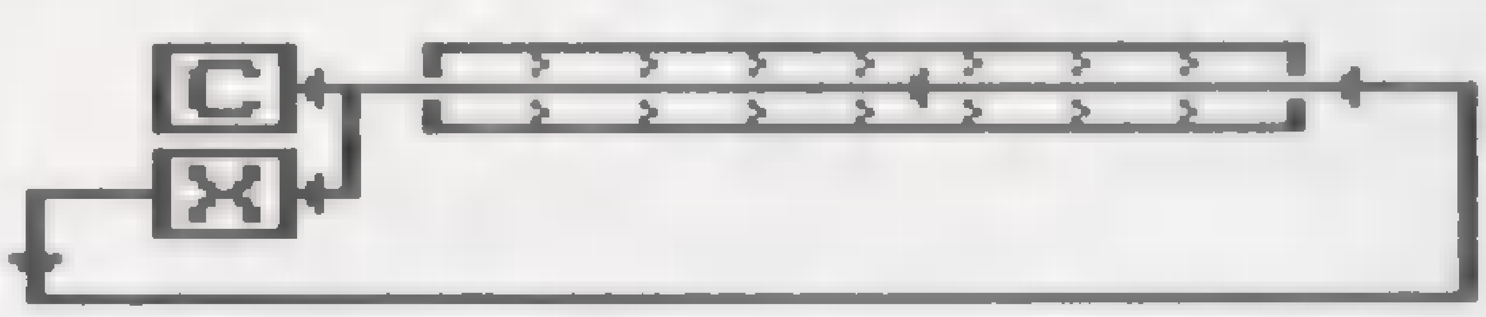
Znaczniki:
X - Ustawiany zgodnie z ostatnim wysuniętym bitem.
N - Ustawiany, gdy najstarszy bit wyniku jest równy jeden. W przeciwnym wypadku zerowany.
Z - Ustawiany, gdy wynikiem operacji jest zero. W przeciwnym wypadku zerowany.
V - Zawsze zerowany.
C - Ustawiany zgodnie z ostatnim wysuniętym bitem.

W rozkazie ROXR można stosować następujące tryby adresowania:

ROXR (An)
ROXR (An)+
ROXR -(An)
ROXR p(An)
ROXR p(An,Rm.W)
ROXR p(An,Rm.L)
ROXR nn.W
ROXR nn.L

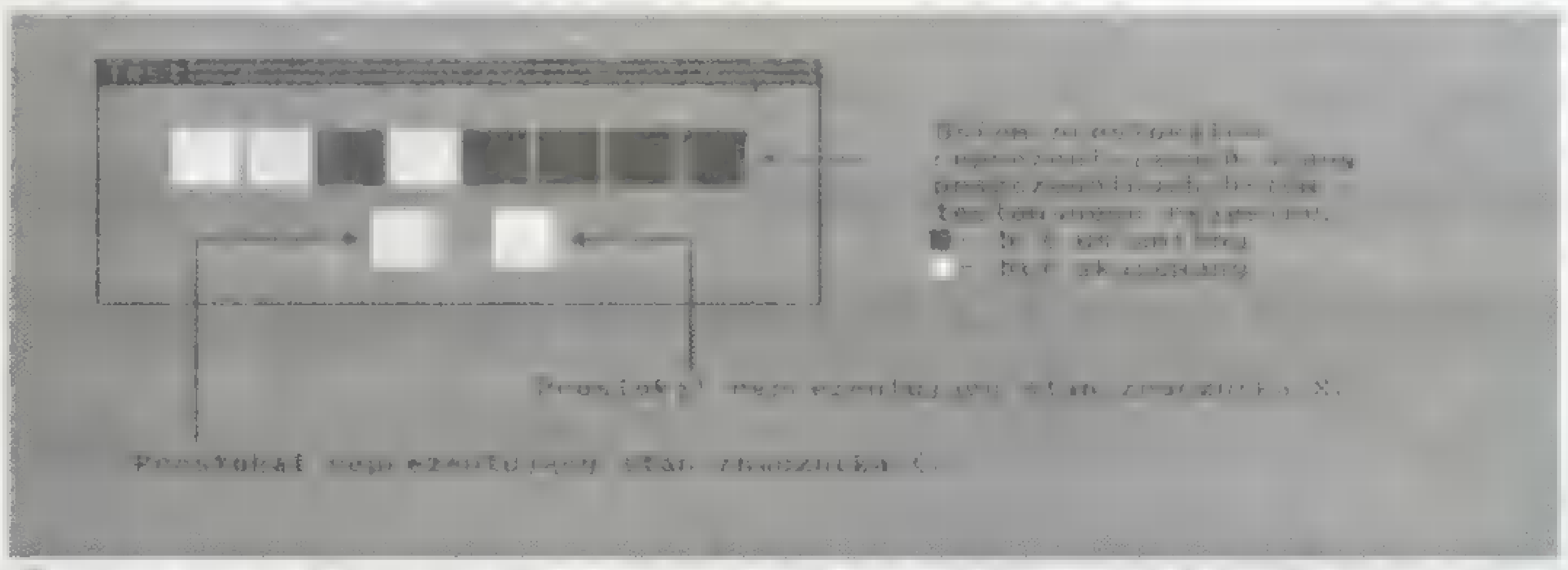
ROXL - Rotate left with extend (Obrót w lewo z rozszerzeniem.)

Zapis: ROXL #n,Dn , ROXL przeznaczenie lub ROXL Dn,Dm
Instrukcja ROXL wykonuje dokładnie te same operacje, co ROXR, tylko że w lewo. ROXL posiada te same ograniczenia w stosowaniu i możliwe tryby adresowania co ROXR. Znaczniki również zachowują się identycznie jak przy ROXR.



Działanie rozkazu ROXL.

To już wszystkie rozkazy z grupy obrotów i przesunięć. Na zakończenie niewielki program (Listing), przy pomocy którego będziecie mogli obejrzeć jak działają wyżej omówione rozkazy. Program rysuje osiem prostokątów przedstawiających bity bajtu, który jest poddawany działaniu testowanego rozkazu oraz dwa prostokąty odpowiadające znacznikom C i X.



Opis okna wyświetlanego przez program z Listingu.

Ci z Was, którzy wpisali program zamieszczony w poprzednim odcinku kursu, zapewne zauważą, że duża część dzisiejszego programu jest identyczna z poprzednim. Nie trzeba więc jeszcze raz wpisywać całości - wystarczy zamienić tylko różniące się fragmenty. Właściwie większość programu została opisana w poprzednim odcinku, więc nie będę tutaj powtarzał dosyć długiego opisu wywoływania funkcji graficznych systemu operacyjnego Amigi. Wspomnę tylko o czterech nowych funkcjach.

Dwie z nich to funkcje z biblioteki intuition.library: WbenchToFront - powodująca ustawienie ekranu workbencha przed innymi ekrananmi i WbenchToBack - chowająca ekran workbencha za pozostałymi ekranami.

Dwie pozostałe to funkcje biblioteki graphics.library: SetAPen - określa, który kolor z palety będzie używany do rysowania oraz RectFill, która rysuje wypełniony prostokąt używając do tego celu aktualnego koloru (ustawionego na przykład funkcją SetAPen).

Dokładniejsze opisy podające jakich parametrów wymagają te funkcje są podane w komentarzach programu.

Po wpisaniu programu z listingu możecie testować działanie różnych rozkazów przesunięć i obrotów. W pierwszej linii za etykietą TestujRozkaz: (move.b #%10000000,d6) możecie wpisać początkową wartość, która będzie obracana, w linii bezpośrednio przed etykietą Loop01: (move.w #8-1,d7) - liczbę obrotów lub przesunięć, jaka ma zostać wykonana, a w linii po etykiecie Loop01: (tej z komentarzem "--- testowany rozkaz ---") - rozkaz, który chcecie przetestować. Rozkaz ten powinien działać na rejestr danych D6 i mieć rozmiar bajtu (np. LSL.B #1,D6, ROR.B #2,D6, ROXL #4,D6, ...).

Wpisując program zapewne zauważyliście tajemniczy, wcześniej nie omawiany, rozkaz BTST. Wiecie już więc, jakim rozkazem zajmiemy się w następnym odcinku kursu. A może już teraz , na podstawie komenterzy w programie potraficie powiedzieć (choćby w przybliżeniu) co ten rozkaz robi? Jeśli nie to na pewno będziecie mogli tego dokonać po lekturze następnego odcinka kursu assemblera. ❑

```
;           ## Listing 1 ##

;       ## procedury exec.library ##
_LVOOpenLibrary equ     -552
_LVOCloseLibrary equ    -414

;       ## procedury intuition.library ##
_LVOOpenWindow      equ     -204
_LVOCloseWindow     equ     -72
_LVOWbenchToFront   equ     -342
_LVOWbenchToBack    equ     -336

wd_RPort            equ     50
WINDOWDRAG          equ     2       ;okno można przesuwać po ekranie
GIMMEZEROZERO       equ     1024    ; współrzędna (0,0) nie będzie
                            ;w lewym lewym górnym rogu okna, ale w lewym
                            ; górnam rogu jego części roboczej (wewnętrznej)
WBENCHSCREEN    equ 1               ; okno na ekranie workbencha

;       ## procedury graphics.library ##
_LVOMove        equ     -240
_LVORectFill    equ     -306
_LVOSetAPen     equ     -342

;       ## procedura dos.library ##
_LVODelay       equ     -198

        section     Program,code    ; poinformowanie assemblera,
                                    ; że teraz będzie program, a nie dane
                                    ; (bez tego też będzie dobrze)

Start:  jmp     Main    ; skok do procedury głównej

;       ## Procedury otwarcia i zamknięcia bibliotek oraz okna ##

OpenGraphics:
        move.l  4.W,a6                  ; baza biblioteki exec
        lea     GraphicsName(pc),a1     ; nazwa biblioteki graphics
        moveq   #0,d0                   ; nr wersji biblioteki
        jsr     _LVOOpenLibrary(a6)     ; otworzenie biblioteki
        move.l  d0,GraphBase            ;zapamiętanie adresu
                                        ; bazowego biblioteki graphics
        rts

OpenIntuition:
        move.l  4.W,a6                  ; baza biblioteki exec
        lea     IntuitionName(pc),a1    ; nazwa biblioteki intuition
        moveq   #0,d0                   ; nr wersji biblioteki
        jsr     _LVOOpenLibrary(a6)     ; otworzenie bibioteki
        move.l  d0,IntuiBase            ; zapamiętanie adresu bazowego
                                        ; biblioteki intuition
        rts

CloseGraphics:
        move.l  4.W,a6                  ; baza biblioteki exec
        move.l  GraphBase(pc),a1        ; baza biblioteki graphics
        jsr     _LVOCloseLibrary(a6)    ; zamknięcie biblioteki
        rts

CloseIntuition:
        move.l  4.W,a6                  ; baza biblioteki exec
        move.l  IntuiBase(pc),a1        ; baza biblioteki intuition
        jsr     _LVOCloseLibrary(a6)    ; zamknięcie biblioteki
        rts

OpenWindow:
        move.l  IntuiBase(pc),a6        ; baza biblioteki intuition
        lea     WindowDef(pc),a0        ; adres definicji okna
        jsr     _LVOOpenWindow(a6)      ; otworzenie okna
        move.l  d0,WindowBase           ; zapamiętanie adresu
                                        ; struktury Window
        rts

CloseWdw:
        move.l  IntuiBase(pc),a6        ; baza biblioteki intuition
        mowe.l  WindowBase(pc),a0       ;adres struktury Window
        jsr     _LVOCloseWindow(a6)     ; zamknięcie okna
        rts

WindowDef: dc.w 200             ; współrzędna x okna
        dc.w    20              ; współrzędna y okna
        dc.w    300             ; szerokość okna
        dc.w    90              ; wysokość okna
        dc.b    0               ; kolor detali w oknie
        dc.b    1               ; kolor wypełnień w oknie
        dc.l    0               ; znaczniki IDCMP
        dc.l    WINDOWDRAG+GIMMEZEROZERO
                        ;znaczniki określające wygląd i typ okna
        dc.l    0       ; adres pierwszej struktury gadget dla tego okna
        dc.l    0
        dc.l    WindowName      ; adres testu będącego nazwą okna
        dc.l    0               ; wskaźnik struktury screen dla ekranu na
                                ;  którym ma się otworzyć okno
                                ; (w tym przypadku równy zeru,
                                ; bo używamy ekranu workbencha
        dc.l    0               ; adres struktury BitMap jeśli chcemy używać
                                ; własnych bitmapów (my nie chcemy)
        dc.w    0               ; minimalna szerokość okna
        dc.w    0               ; minimalna wysokość okna
        dc.w    240             ; maksymalna szerokość okna
        dc.w    200             ; maksymalna wysokość okna
        dc.w    WBENCHSCREEN    ; typ okna
                                ; (w naszym przypadku okno
```

```
                                ; na ekranie workbencha)

WindowName:     dc.b  'Test',0                  ; nazwa okna
GraphicsName:   dc.b  'graphics.library',0      ; nazwa biblioteki graphics
IntuitionName:  dc.b  'intuition.library',0     ; nazwa  biblioteki intuition
DosName:        dc.b  'dos.library',0           ; nazwa biblioteki dos
GraphBase:      dc.l  0                         ; miejsce przechowywania bazy
                                                ; graphics.library
IntuiBase:      dc.l  0                         ; miejsce przechowywania bazy
intuition.library WindowBase:   dc.l  0         ; miejsce przechowywania adresu
                                                ; struktury Window dla okienka

WBtoFront:
    move.l  IntuiBase(pc),a6
    jsr     _LVOWbenchToFront(a6)
    rts

WBtoBack:
    move.l  IntuiBase(pc),a6
    jsr     _LVOWbenchToBack(a6)
    rts

Wait:
    move.l  4.W,a6                  ; baza biblioteki exec
    lea     DosName(pc),a1          ; nazwa biblioteki dos
    moveq   #0,d0                   ; nr wersji biblioteki
    jsr     _LVOOpenLibrary(a6)     ; otworzenie biblioteki
    move.l  d0,a6                   ; adres bazowy biblioteki dos do a6
    move.l  #1*50,d1                ; ilość 1/50 sekundy do odczekania
    jsr     _LVODelay(a6)           ; wywołanie procedury czekającej
    move.l  a6,a1                   ; baza biblioteki dos do a1
    move.l  4.W,a6                  ; bata biblioteki exec
    jsr     _LVOCloseLibrary(a6)    ; zamknięcie biblioteki dos
    rts

ColorSet:                           ; d0 - numer koloru (0-3)
    movem.l d0-a6,-(sp)  ; zapamiętanie wszystkich rejestrów na stosie
    move.l  GraphBase(pc),a6        ; baza biblioteki graphics
    move.l  WindowBase(pc),a1       ; adres stuktury Window
    move.l  wd_RPort(a1),a1         ; adres struktury RastPort
    jsr     _LVOSetApen(a6)         ; wywołanie funkcji SetAPen
            ; ustawiającej numer koloru używanego przy rysowaniu
    movem.l (sp)+,d0-a6             ; zdjęcie wszystkich rejestrów ze stosu
    rts

FillRectangle:                      ; d0 - x lewego,górnego rogu
                                    ; d1 - y lewego,górnego rogu
                                    ; d2 - x prawego,dolnego rogu
                                    ; d3 - y prawgo,dolnego rogu
    movem.l d0-a6,-(sp)  ; zapamiętanie wszystkich rejestrów na stosie
    move.l  GraphBase(pc),a6        ; baza biblioteki graphics
    move.l  WindowBase(pc),a1       ; adres stuktury Window
    move.l  wd_RPort(a1),a1         ; adres struktury RastPort
    jsr     _LVORectFill(a6)        ; wywołanie funkcji RectFill
                                    ; rysującej wypełniony prostokąt
    movem.l (sp)+,d0-a6             ; zdjęcie wszystkich rejestrów ze stosu
    rts


;   ## Program główny ## Main:
    bsr     OpenGraphics    ; wywołanie procedury otwierającej
                            ; bibliotekę graphics
    beq     NoGraphics      ; skok jeśli błąd otwarcia graphics
    bsr     OpenIntuition   ; wywołanie procedury otwierającej
                            ; bibliotekę intuition
    beq     NoIntuition     ; skok jeśli błąd otwarcia intuition
    bsr     WBtoFront       ; ustawienie ekranu workbencha
                            ; przed innymi ekranami
    bsr     OpenWindow      ; wywołanie procedury otwierającej okno
    beq     NoWindow        ; skok gdy błą otwarcia okna
    bsr     TestujRozkaz
    bsr     CloseWdw        ; wywołanie procedury NoWindow:
                            ; zamykającej okno
    bsr     WBtoBack        ; ustawienie ekranu workbencha
                            ; za innymi ekranami
    bsr     CloseIntuition  ; wywołanie procedury zamykającej
                            ; NoIntuition:
                            ; bibliotekę intuition
    bsr     CloseGraphics   ; wywołanie procedury zamykającej
                            ; NoGraphics:
                            ; bibliotekę graphics
    rts                     ; powrót z programu


TestujRozkaz:
    move.b  #%10000000,d6   ; początkowa wartość
    bsr     RysujRejestr    ; narysuj rejestr D0 przed przesunięciem
    bsr     Wait            ; odczekanie 1 sekundy
    moveq   #8-1,d7         ; dokonamy ośmiu przesunięć Loop01:
    lsr.b   #1,d6           ;  testowany rozkaz --
    bsr     RysujRejestr    ; narysuj bity rejestru D0; po przesunięciu
    bsr     Wait            ; odczekanie 1 sekundy

    dbf     d7,Loop01
    rts

RysujRejestr:
    move    SR,d5           ; pobranie rejestru statusowego
                            ; i zapamiętanie go w rejestrze D5
    move.w  #110,d0         ; x bitu C
    move.w  #45,d1          ; y bitu C
    btst    #0,d5           ; badanie stanu bitu C
    bsr     RysujJedenBit

    move.w  #160,d0         ; x bitu X
    move.w  #45,d1          ; y bitu Y
    btst    #4,d5           ; badanie stanu bitu X
    bsr     RysujJedenBit

    move.w  #28,d0          ; x pierwszego "bitu"
    move.w  #15,d1          ; y pierwszego "bitu"
    btst    #7,d6           ; test 7 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #6,d6           ; test 6 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #5,d6           ; test 5 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #4,d6           ; test 4 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #3,d6           ; test 3 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #2,d6           ; test 2 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #1,d6           ; test 1 bitu
    bsr     RysujJedenBit
    add.w   #30,d0          ; obliczenie x kolejnego bitu
    btst    #0,d6           ; test bitu 0
    bsr     RysujJedenBit

    rts

RysujJedenBit:
    beq     bitZerowy       ; skok, gdy bit zapalony
                            ;BitNieZerowy:
    move.l  d0,-(sp)        ; zapamiętanie rejestru D0
    move.w  #1,d0           ; gdy bit zerowy, to rysuj
                            ; kolorem numer 1
    bra     RysujJedenBit01 BitZerowy:
    move.l  d0,-(sp)
    move.w  #2,d0           ; gdy bit nie zerowy, to rysuj
                            ; kolorem numer 2 RysujJedenBit01:
    bsr     ColorSet        ; ustawienie wybranego koloru
    move.l  (sp)+,d0
    move.w  d0,d2           ; skopiowanie lewego,górnego x
                            ; prostokąta do D2
    move.w  d1,d3           ; skopiowanie lewego,górnego y
                            ; prostokąta do D3
    add.w   #26,d2          ; dodanie szerokości prostokąta
                ; i uzyskanie w ten sposób x prawego,dolnego rogu
    add.w   #20,d3          ; dodanie wysokości prostokąta
                ; i uzyskanie w ten sposób y prawego,dolnego rogu
    bsr     FillRectangle   ; narysowanie wypełnionego prostokąta
    rts
```

KONKURS KONKURS KONKURS

Maciej Klimkiewicz

Zadanie 1

Ci, którzy myśleli, że to zadanie jest zbyt proste, żeby się nad nim zastanowić popełnili błąd. Wbrew pozorom poprawna odpowiedź nie brzmi 10 KB, ale 5 KB, co łatwo można sprawdzić układając układ równań:

program + obrazek = 310
obrazek = 300 + program

który po podstawieniu 2-go równania do 1-go przyjmuje postać:

program + 300 + program = 310

Jego rozwiązaniem jest: program = 5

Zadanie 2

To zadanie okazało się dosyć trudne, chociaż każdy "komputerowiec" spotyka się z przedstawionym ciągiem liter na co dzień. Jeżeli nie potrafiliście rozwiązać tego zadania - spróbujcie jeszcze raz, ale czytając ciąg od tyłu. Jeżeli i to Wam nic nie pomogło - spójrzcie na środkowy rząd klawiatury, a przekonacie się, że brakującą literą ciągu jest "G".

Zadanie 3

Z tym zadaniem nie było większych trudności. Wystarczyło wpaść na pomysł, że do częściowo zapełnionych wiader można dolać płynu. Istnieje wiele rozwiązań tego zadania - ja podaję jedno z nich:

wiadro 1 pojemność 8l	wiadro 2 pojemność 5l	wiadro 3 pojemność 3l
8	0	0
3	5	0
3	2	3
6	2	0
6	0	2
1	5	2
1	4	3
4	4	0

Zadanie 4

Sukces rozwiązania tego zadania zależy od sposobu zapisania danych i później, od wyciągnięcia na ich podstawie odpowiednich wniosków. Można to zrobić w następujący sposób:

na Bajtka naskarżył:	Pajtek
na Majtka naskarżył:	Kajtek
na Fajtka naskarżyli:	Majtek i Kajtek
na Pajtka naskarżył:	Bajtek
na Kajtka naskarżyli:	Bajtek, Fajtek i Pajtek
na Rajtka naskarżyli:	Majtek i Fajtek

Z zadania wiemy, że w sumie były 4 prawidłowe skargi i 6 nieprawidłowych, a więc musimy wybrać taką parę chłopaków, żeby suma wniesionych na nich skarg wynosiła 4. Do tego idealnie pasowaliby Fajtek i Rajtek, ale na obu z nich naskarżył Majtek - a wiadomo, że żaden z nich nie wskazał obu sprawców prawidłowo. Musimy zatem wybrać inną parę. I w tym momencie już widać, że stworzenie innej pary możliwe jest tylko z Kajtkiem - a więc jest on na pewno sprawcą. Poszukajmy zatem jego wspólnika. Rozumując podobnie jak przed chwilą, możemy z grona podejrzanych wykluczyć te osoby, na które naskarżyli: Bajtek, Fajtek i Majtek, czyli na pewno sprawcami nie byli: Pajtek, Rajtek i Bajtek. Wykluczyć możemy również Fajtka, ponieważ suma skarg Fajtka i Kajtka nie wynosi 4. Pozostał tylko Majtek. Zatem poprawna odpowiedź brzmi: sprawcami byli Kajtek i Majtek, albo jak kto woli Majtek i Kajtek.

Zadanie 5

Zadanie to okazało się dość trudne. Stworzenie trójkąta równobocznego nie jest możliwe o żadnej godzinie. Jednak sytuacja temu najbliższa nastąpi o 09:05:27.3. Jedną z wielu możliwych dróg prowadzących do tego rozwiązania jest przejście z systemu dziesiętnego na dwunastkowy (bo jest dwanaście godzin na tarczy). W systemie tym można zapisać dowolną godzinę przy pomocy liczby trójcyfrowej: gms, gdzie g - oznacza godziny, m - minuty-dwunastkowe, s - sekundy-dwunastkowe. Trzeba tylko pamiętać, że 12 minut-dwunastkowych odpowiada 1 godzinie. Podobnie jest z sekundami.

Wiadomo, że suma kątów pomiędzy wskazówkami wynosi 180 stopni, albo inaczej rzecz ujmując, dwanaście godzin. Na tej podstawie możemy napisać pierwsze równanie. Wiadomo również, że wszystkie kąty mają być sobie równe (trójkąt równoboczny), czyli pomiędzy sąsiadującymi wskazówkami muszą być 4 godziny-dwunastkowe. W ten sposób otrzymaliśmy drugie równanie. Mając dwa równania z trzema niewiadomymi szukamy ekstremów powstałej funkcji. Na koniec trzeba pamiętać, żeby wynik-dwunastkowy przekształcić na rzeczywisty.

Zadanie 6

Wiadomo, że taboret ma trzy nogi, a krzesło cztery i że siedzą na nich ludzie, którzy też mają nogi - zatem na każdy taboret przypada pięć nóg, a na krzesło sześć. Czyli mamy równanie:

$$t * 5 + k * 6 = 39$$

Jeżeli t-taborety i k-krzesła mają być liczbami całkowitymi dodatnimi, to istnieje tylko jedno rozwiązanie: w pokoju były trzy taborety i cztery krzesła.

W tym miejscu zaczynają się zadania, które były przewidziane do rozwiązania przy pomocy komputera. Ze względu na to, że nie ma miejsca na drukowanie przykładowych programów, podam tylko gotowe rozwiązania.

Zadanie 7

Suma liczb od 1 do 1000 wynosi: 500500

Zadanie 8

Tekst po rozszyfrowaniu brzmi:
"Celem programowania jest nie program, lecz wynik obliczenia."

Zadanie 9

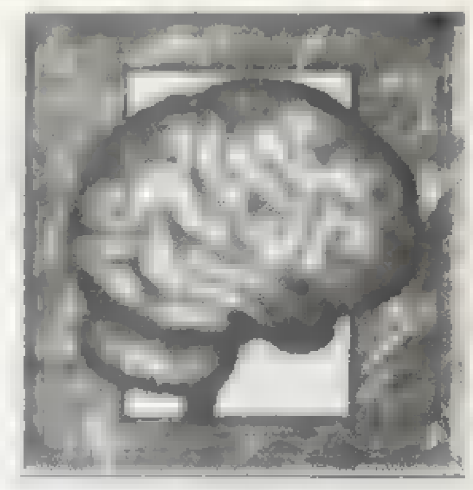
Drugą parzystocyfrową liczbą palindromiczną jest:

637832238736 = 798644"

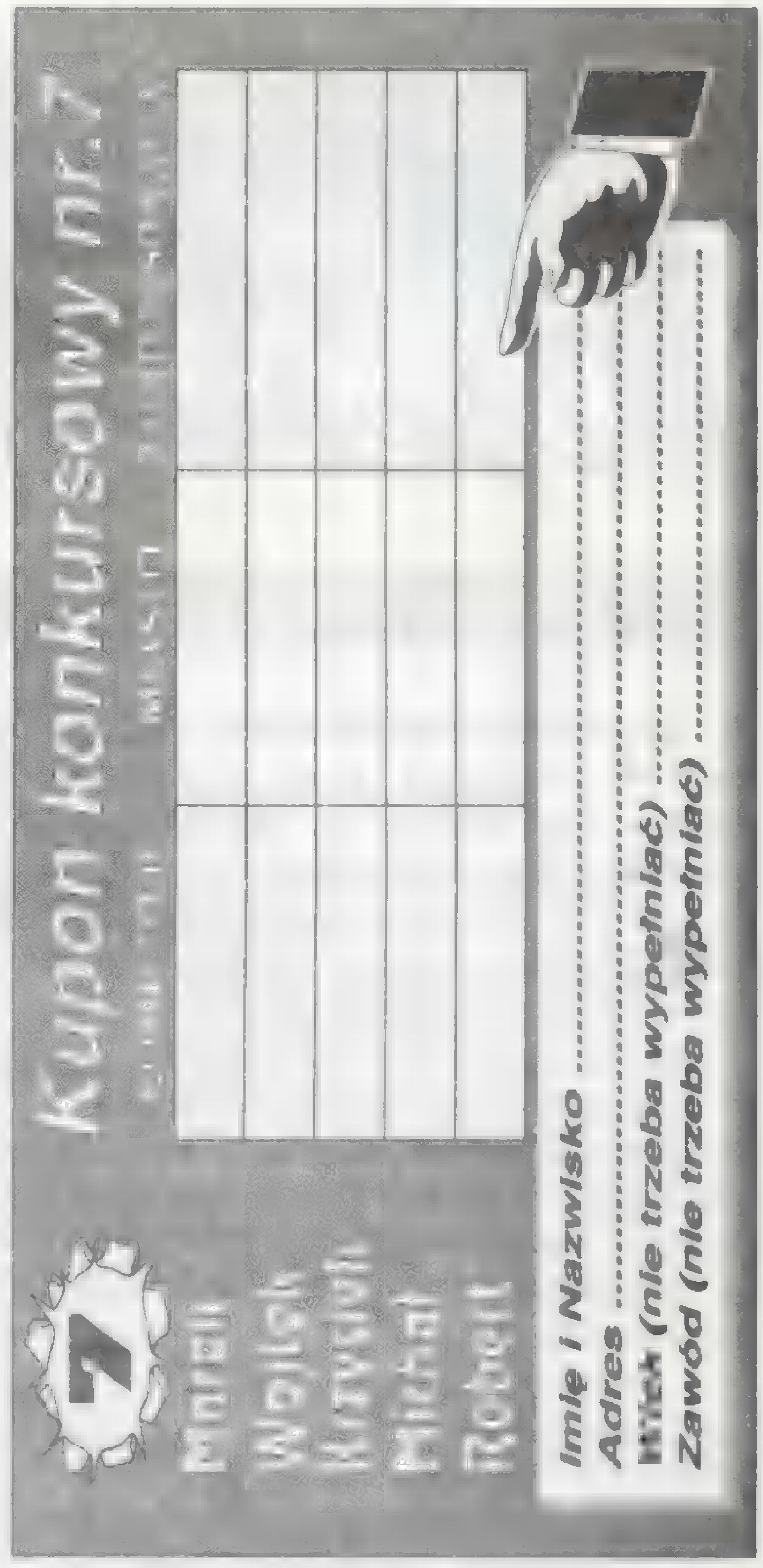
Poprawne rozwiązania przysłali:

Szymon Jończy - Pszczyna
Sławomir Roman - Olsztyn
Tomasz Wiśniewski - Gniwkowo

Nagrody prześlemy pocztą.



	A500	A1000	A4000/30	A600	A3000	GRY	DEMA	VIDEO	GRAFIKA	DTP	POZNAŃ	GDAŃSK	KRAKÓW	BYDGOSZCZ	WROCŁAW
MAREK															
WOJTEK															
KRZYSIEK															
MICHAŁ															
ROBERT															
POZNAŃ															
KRAKÓW															
BYDGOSZCZ															
GDAŃSK															
WROCŁAW															
GRY															
DEMA															
VIDEO															
GRAFIKA															
DTP															



Tym razem tylko jedno zadanie. Poniżej przedstawione są fragmenty pewnego sprawozdania. Na podstawie uzyskanych z niego informacji należy wypełnić kupon konkursowy i przesłać go na adres redakcji:

"Amigowiec"
dział RUSZ GŁOWĄ
ul. Świętojańska 2/7
85-017 Bydgoszcz.

Wśród nadesłanych poprawnych odpowiedzi rozlosujemy nagrody. Rozwiązanie w numerze 3/94.

Fragmenty sprawozdania ze zlotu Amigowców w Bydgoszczy

Zlot Amigowców odbył się 30 lutego 1993 roku w Bydgoszczy. Jego organizatorem był 24 letni Wojtek - student medycyny.

Jednym z uczestników zlotu był Michał Kokosek z Poznania, który przyjechał na zlot, żeby wymienić swoją Amigę na Amigę 3000. Niestety, nie znalazł nikogo chętnego do zamiany i musiał wrócić do domu ze starym sprzętem.

Wśród wielkich osobistości Amigowskiej sceny można było spotkać niejakiego Koczara z Krakowa - eksperta od animacji wideo, który przywiózł ze sobą swoje 32-bitowe cacko, na którym prezentował animacje.

Na spotkaniu obecny był również wspaniały grafik Kwarkowski z Wrocławia, który nie miał Amigi 600.

W drugim dniu zlotu odyła się konferencja dotycząca DTP. Prowadzący ją chłopak demonstrował na swojej Amidze różne techniki składu. Ciekawe jest to, że posiadając mało pamięci, nie odczuwał jej braku. Zapewne spowodowane było to zainstalowaną pamięcią wirtualną.

Zdziwienie budził najstarszy - jeszcze działający model Amigi, który przywiózł ze sobą Krzysztof. Zapytany przez nas, czy na swoim sprzęcie może uruchamiać nowe gry, odpowiedział, że ten dział oprogramowania go nie interesuje.

Marek, który miał Amigę 600, nie grał na niej.

PISMA Pisemka . . .

Tomasz Łobuda

Czy do kontrolera firmy "Protar" można podłączyć dysk twardy od IBMa? Czy dałoby się zaadaptować dysk twardy od laptopa?

Dysk twardy sam w sobie jest urządzeniem dość bezmyślnym. To właśnie kontroler nadaje mu "osobowość". Inaczej mówiąc, taki sam dysk twardy może znaleźć się zarówno w IBMie, jak i w naszej Amidze. Trzeba jednak zaznaczyć, że kontroler znajdujący się w "Protarze" jest standardu SCSI, ■ więc wymaga odpowiedniego dla niego dysku twardego (AT-BUS odpada).

Dyski twarde od laptopów (o wielkościach 1.8" lub 2.5") są w przeważającej części standardu AT-BUS. Jeżeli już udałoby się znaleźć tego typu dysk w standarcie SCSI, wówczas cały problem polegałby na wykonaniu odpowiedniej "przejściówki" (coś na kształt złączki opisywanej w artykule "Twardy dysk dla Amigi" w "Amigowcu" nr 7-8/93) oraz na stabilnym zamocowaniu w obudowie "Protara".

Czy do A1200 można podłączyć stację dysków HD? Czy można zastosować do tego stację HD od IBMa? Jak wygląda współpraca z taką stacją (formatowanie, kopiowanie, itd)?

Można. W pierwotnych planach konstruktorów stacja taka miała być wyposażeniem standardowym. Niestety. Okazało się, że firma, z którą współpracował koncern Commodore produkowała zbyt wysokie stacje HD. Nie mieściły się one w tak strasznie upakowanym komputerze jakim jest A1200. Wydawać by się mogło, że poprawienie takiego drobiazgu nie jest specjalnie trudne. Ponieważ czas naglił, a prawie cały proces technologiczny już był gotowy, zdecydowano się na instalację starych stacji dysków. Jednak najważniejsza część pierwotnych zamiarów pozostała. Otóż A1200 posiada kontroler umożliwiający obsługę stacji o podwyższonej gęstości. Stację taką można podłączyć wewnętrznie (stacja 3,5" - wiąże się to z drobnymi przeróbkami wewnątrz komputera) lub wyprowadzić przewody szyny danych na zewnątrz i wówczas spokojnie "przypasować" napęd HD (3,5" lub 5,25"). Stacji takich można szukać w serwisach lub na giełdach (nieczęsto, ale można spotkać napędy od A3000 lub A4000). Jeżeli chodzi o drugie pytanie, możliwe jest wykorzystanie stacji HD IBMa do Amigi. Stacja taka była opisywana w poprzednim AMIGOWCU (str. 54). Jest to oczywiście model zewnętrzny, który do poprawnej pracy z Workbenchem wymaga sterownika w postaci dodatkowego programu. Ta cecha odróżnia napęd IBMowski od Amigowskiego. (UWAGA: IBMowska stacja HD zaimplementowana do Amigi bez odpowiedniego oprogramowania jest bezużyteczna.) Natomiast współpraca z oryginalnym napędem amigowskim jest dla systemu 3.0 czymś najzupełniej normalnym i może się odbywać bezpośrednio z Workbencha, w taki sam sposób jak ze zwykłymi stacjami.

Czy istnieje możliwość podłączenia do A2000D stacji HD?

Taka możliwość istnieje. Wymaga to zainstalowania w komputerze nowego kontrolera potrafiącego obsługiwać napędy dysków HD. Stację taką można wmontować w miejsce normalnej stacji 3,5" (mogą być potrzebne pewne przeróbki) lub zaopatrzyć się w jej specjalną, zewnętrzną wersję. Niestety, w Polsce dla większości dystrybutorów liczą się jedynie małe Amigi: A500 i A600. Stąd zakup jakichkolwiek akcesoriów do innych modeli jest dość karkołomną zabawą. Jako potencjalne źródło zaopatrzenia się w kontroler dysków HD można jedynie wskazać warszawskie sklepy i giełdy (lub wyjazd za granicę).

Jak wygląda wymiana danych pomiędzy Macintoshem ■ Amigą? Czy istnieje coś takiego jak "DOS-2-DOS" (np. "MAC-2-DOS") i jak wygląda wówczas sprawa zapisu danych?

Bez uciekania się do stosowania specjalnych programów lub rozszerzeń sprzętowych można wymieniać z MACiem pliki tekstowe (w standarcie ASCII) oraz grafikę: wektorową (EPS) i bitową (skompresowaną-JPEG, IBMopodobną - GIF,TIF oraz amigowską - IFF). Do przenoszenia wykorzystuje się wspólną dla Amigi i Macintosha możliwość zapisu i odczytu dyskietek w IBMowskim systemie MS-DOS. Załóżmy, że chcemy przenieść długi plik graficzny. Na Amidze musimy najpierw uruchomić program "PC0", znajdujący się w szufladzie "DOSDrivers" i sformatować dyskietkę jako MS-DOS (z poziomu Workbencha lub z pomocą programów zarządzających np. "Directory Opus"). Następnie na tak przygotowany dysk kopiujemy interesującą nas grafikę. Jeżeli nie zmieści się ona na standardowej dyskietce, wówczas możemy użyć programu "Split&Join" (dołączanego do programu "Art Department Pro" i będącego uniwersalnym narzędziem do przenoszenia grafiki między IBMem, Amigą i Macintoshem) mającego możliwość rozbicia rysunku na kilka dysków (do "poskładania" którego używa się "Split&Joina" w wersji Macintoshowej). Na Mac-u natomiast można odczytać dane z dyskietki MS-DOS-owej po uruchomieniu programów "Soft-PC" lub "PC-Exchange". Do wymiany danych mogą służyć również sprzętowe emulatory Macintosha takie jak: Emplant czy A-MAX . Jest to jednak bardzo kosztowna inwestycja. (Po dokładniejsze

informacje o Emplancie odsyłamy do numeru 9-10/93, str. 6 - przyp. red.)

Jeżeli chodzi o drugie pytanie: rzeczywiście istnieje program "MAC-2-DOS" (spełnia on analogiczną rolę jak "DOS-2-DOS"), jednak dla prawidłowej pracy wymaga stosowania specjalnej, zazwyczaj oryginalnej, Macintoshowej stacji dysków. Wynika to ze specyfiki pracy stacji Macintosha, a konkretnie ze zmiennej szybkości obrotów nośnika z jaką pracuje napęd MACa.

Czy możliwe jest rozszerzenie pamięci Chip Amigi 2000D do 2 MB?

W przypadku wymienionej Amigi jest to możliwe. Moduł rozszerzenia składa się z płytki zawierającej nową wersję Agnusa - FatAgnus (stara mogła adresować tylko 1MB) i z 2 MB rozszerzenia pamięci. Rozszerzenie tego typu wymaga specjalnego sposobu montowania (usunięcia z płyty starego Agnusa), o czym powinna informować dołączona instrukcja. Cena ok. 350 DM.

Czy istnieje odpowiednik napędu CD (A570) do Amigi 2000D?

Bezpośredniego odpowiednika tego napędu nie ma. Przyczyna jest bardzo prosta: firma Commodore nie produkuje żadnej stacji CD do A2000, 3000 i 4000. Można natomiast zaopatrzyć się w "niefirmowy" CD-ROM wykorzystujący znany z CDTV system ISO 9660/High-Sierra lub inne multisystemowe jak np. Sony czy Toshiba. Powyższe napędy są podłączane do komputera poprzez kontroler SCSI wewnętrznie (w miejsce stacji 5.25") lub zewnętrznie (w osobnej obudowie). Ich obsługa z poziomu systemu jest możliwa tylko z pomocą odpowiedniego oprogramowania. Cena bez oprogramowania ok. 900 DM.

Czy istnieją programy typu CAD na Amidze? Jeśli tak, to jakie i czy istnieje możliwość przenoszenia danych z AutoCADa?

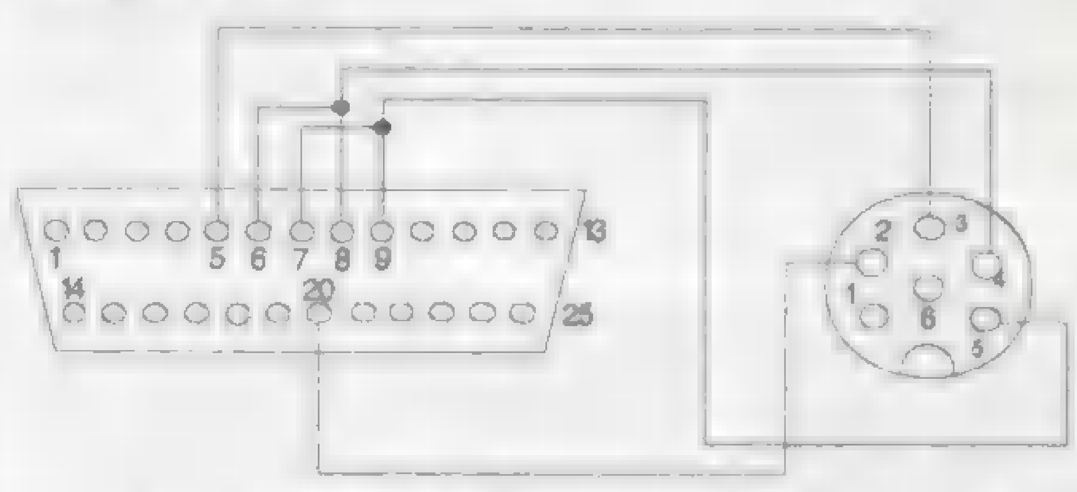
Programy typu CAD istnieją na Amidze od dawna i jest ich nawet sporo. Do najprostszych należą: "Draw 2000" i "Intro CAD". Oba rozpoczynają pracę od minimalnej konfiguracji sprzętowej - wystarcza im Amiga 500 z 512 kB RAM. Dalej lokuje się cała grupa "średniaków", do której należą między innymi: "X-CAD" oraz "Dynamic CAD". Według nie tylko mojej oceny najlepszymi programami tego typu są: "Dyna CAD" i "Maxon CAD". Pierwszy z nich oferuje olbrzymie wręcz możliwości. Od prostych opcji rysunkowych, aż po rzutowanie perspektywiczne. Odbywa się to kosztem dużej "pamięciożerności". "Maxon CAD" natomiast zadawala się, jak na tego typu program, minimalną ilością RAMu, a mniejsze możliwości skutecznie nadrabia niesamowitą szybkością pracy.

Jeżeli chodzi o przenoszenie danych z "AutoCADa", to sprawa z pozoru wygląda prosto. Od "X-CADa" w górę wszystkie amigowe programy typu CAD potrafią obsługiwać dane zapisane w standardzie DXF. Standard ten narodził się wraz z pierwszymi wersjami "AutoCADa". Później wielokrotnie go ulepszano, a kiedy okazało się, że więcej się z niego nie wyciśnie, IBMowcy wymyślili coś nowego i nazwali to DWG. To "kombinowanie" niekorzystnie odbiło się na zgodnościach starych DXF-ów z nowymi. Wszystko to mocno utrudnia łatwe z pozoru przenoszenie danych między "AutoCADem", a CADami amigowymi. Na pocieszenie pragnę zawiadomić, że wkrótce na łamach "Amigowca" pojawi się obszerny artykuł dotyczący wymiany danych między Amigą, IBMem i Macintoshem.

Jak podłączyć stację 1541 do Amigi poprzez port RS 232? Czy potrzebna jest do tego odpowiednia wtyczka? Jak ją wykonać?

Sądzę, że zaszło tu drobne nieporozumienie. Oryginalną stację 1541 (normalnie współpracującą jedynie z komputerami Commodore C64 lub C128) da się podłączyć do Amigi jedynie przez port "Parallel". Spotkałem się natomiast z informacją, o możliwości bezpośredniego połączenia komputerów: Amiga i C64, właśnie za pomocą portu RS 232. Wymaga to zastosowania odpowiedniego złącza, a także połączenie takie musi być obsługiwane za pomocą specjalnie napisanych procedur do transmisji danych.

Jeżeli chodzi o wspomniane połączenie stacji 1541 przez port "Parallel" to sprawa przedstawia się stosunkowo prosto. Należy zaopatrzyć się w następujące elementy: złącze typu DIN, standardowe (25 bolcowe) złącze typu D oraz czterożyłowy przewód. Schemat połączenia został przedstawiony na rysunku.



Parallel port (Amiga) Serial port (C64)



Public Domain!

Nowy Rok, nowy Amigowiec i nowa dyskietka PD. A oto spis programów, które na niej znajdziesz.

1. ROM Boot Patch 1.2

Program ten likwiduje błąd systemu 2.0 polegający na tym, że po resecie zawartość SLOW-RAMu jest uszkadzana w kilku miejscach (w zależności od tego, ile masz pamięci - od dwóch do sześciu). Zjawisko to mogą zaobserwować tylko posiadacze przełącznika Kickstartów w A500, gdyż posiadacze Amig 500+ i 600 nie mają po prostu SLOW-RAMu. Z tej właśnie przyczyny program ten NIE BĘDZIE działał z systemami 1.2, 1.3 i 3.0. Jest przeznaczony WYŁĄCZNIE dla użytkowników systemu 2.0, testowany był pod jego wersją 2.04.

Wspomniany wyżej błąd praktycznie uniemożliwia pracę z takimi programami rezydentnymi, jak RAD czy RRD - ich zawartość jest po resecie uszkodzona.

Pamięć jest uszkadzana w stałych miejscach:

$C3F09A i $C3F09B,
$C7F09A i $C7F09B,
$CBF09A i $CBF09B,
$CFF09A i $CFF09B,
$D3F09A i $D3F09B,
$D7F09A i $D7F09B.

Dzięki temu możliwe jest usunięcie tego błędu. I to właśnie robi program ROMBootPatch.

Program ten możesz uruchomić zarówno z Workbencha, jak i z CLI.

Uruchomienie z CLI

Możliwe jest podanie dwóch parametrów:

PATCHRAD/S,MOUNTRAD/S

Podanie pierwszego z nich powoduje zmianę sposobu działania programu, tzn. dokonane zmiany będą kompatybilne z RADem.

Podanie drugiego parametru zostanie przez program zauważone tylko wtedy, gdy podany zostanie pierwszy. Powoduje on wywołanie komendy "Mount RAD:". Może Ci to uprościć czynności, tzn. przez wywołanie "ROMBootPatch PatchRad MountRad" zostanie od razu utworzony poprawiony "RAD:".

Uruchomienie z Workbencha

Parametry identyczne jak powyżej podaje się w ToolType ikony (menu Icons/-Information).

Możliwe jest jeszcze podanie parametru:

WINDOW=CON:x/y/szerokość /wysokość/nazwa_okna

Spowoduje on otworzenie okna-konsoli będącej plikiem I/O dla programu (w nim będą wyświetlane komunikaty). Bez podania tego ToolType program również będzie działał, ale nie dowiesz się o ewentualnych problemach itp.

Uruchomienie programu bez podania 'PatchRad'

W takim przypadku program działa na następującej zasadzie: skoro pamięć jest uszkadzana zawsze w tych samych miejscach, to nie można pozwolić, aby jakiś rezydentny program został zainstalowany w tym miejscu. Trzeba tę pamięć po prostu zająć. Inny program nie będzie mógł z niej skorzystać, czyli wszystko będzie OK. Zajmowane jest najwyżej 48 bajtów pamięci (ilość zagrożonych obszarów * 8). Przy takim uruchomieniu możliwe są następujące komunikaty:

1. "Pamięć pod adresem _adres_ jest już zajęta!". Komunikaty takie ujrzysz przy dwukrotnym uruchomieniu ROMBootPatcha (bez resetu między tymi uruchomieniami). Jeżeli jednak jest to pierwsze uruchomienie, to sprawa jest nieco poważniejsza: Jakiś program już zajął dany obszar pamięci.

2. "Wszystkie zagrożone obszary pamięci zabezpieczone.". Działania programu zakończyły się całkowitym sukcesem.

3. "Twoja AMIGA nie posiada SLOW-RAMu!". Nie ma co poprawiać.

ROMBootPatch należy, przy tym sposobie uruchamiania, uruchamiać koniecznie PRZED instalacją programu rezydentnego! Nie jest niezbędne uruchamianie go po każdym resecie komputera (program rezydentny instaluje się przecież tylko raz, potem już siedzi sobie w pamięci). Istnieje jeden wyjątek: RRD. Przy korzystaniu z tego programu należy ROMBootPatcha uruchamiać każdorazowo po resecie, gdyż RRD zajmuje pamięć w sposób dynamiczny. Należy się zastanowić, czy warto zainstalować ROMBootPatcha na każdym dysku. Jak wiesz, dzieli on pamięć na kilka kawałków, a niektóre programy lubią mieć jeden, duży obszar pamięci. Tak więc przy korzystaniu np. z programu kompresującego czy emulatora IBMa, lepiej ROMBootPatcha nie używać. Szczególnym przypadkiem jest RAD, ale o tym w następnym punkcie.

Uruchomienie z podaniem 'PatchRad'

Podczas testów programu okazało się, że po użyciu ROMBootPatcha niemożliwe jest zamontowanie RADu. Dzieje się to z dość głupiego powodu - cały RAD MUSI się znajdować w jednym kawałku pamięci. Ponieważ po użyciu ROMBootPatcha pamięć SLOW-RAM jest podzielona na 7 kawałków, RADu nie da się zamontować (ewentualnie możesz zamontować kilka mniejszych RADów, ale to nie jest sensowne rozwiązanie.). Tak więc trzeba było wymyślić inny sposób zabezpieczenia RADu. I oto jego zasada działania: Program wylicza, które bloki zostaną uszkodzone. Robi to na podstawie analizy prywatnych danych "ramdrive.device". Kiedy już pozna numery za

grożonych bloków, odnotowuje w bitmapie RADu, że dane bloki są zajęte. Po tej operacji system nie będzie już mógł na to miejsce nagrać żadnych innych danych. Program ten powoduje stratę zmiennej ilości miejsca w RADzie: od 0 (nie ma potrzeby niczego poprawiać, bo RAD nie znajduje się w zagrożonym obszarze) do 3 KB (6 bloków jest zagrożonych). ROMBootPatch wystarczy uruchomić tylko raz, zaraz PO zamontowaniu RADu (lub podając parametr MountRad). NIE WOLNO później wykonywać na RADzie operacji w stylu formatowania czy diskcopy'owanie - gdyż zniweczy to cały efekt. Niepowodzeniem zakończy się także przypuszczalnie próba jego optymizacji. Tak przygotowanego RADu lepiej nie poddawać działaniu programów DiskDoctoro-podobnych. Przy takim trybie uruchomienia możliwe są następujące komunikaty:

1. "Brak pamięci!" - bez komentarza.

2. "'RAD: nie jest zainstalowany!" - bez komentarza.

3. "Nie ma potrzeby dokonywania korękt dla tego "RAD'u." - RAD jest zainstalowany tak szczęśliwie, że nie koliduje z zagrożonymi obszarami pamięci, nie ma więc czego poprawiać.

4. "Blok _nr_bloku_ jest już zajęty!". Taki komunikat spotka Cię przy dwukrotnym uruchomieniu programu ROMBootPatch. Jeśli jednak jest to pierwsze jego uruchomienie, to znaczy, że coś już w zagrożonym bloku siedzi, a po resecie zostanie uszkodzone!

5. "Wyłączyłem z użytku blok _nr_bloku_". Zagrożony blok został wyłączony. Numer nie jest w zasadzie ważny dla użytkownika, służy bardziej autorowi - do wykrywania ewentualnych błędów w programie.

6. "Wszystkie zagrożone bloki zabezpieczone.". Działania programu zakończyły się sukcesem.

UWAGA!!!
RAD, który chcesz poddać korekcie, musi się cechować następującymi rzeczami (w przeciwnym wypadku mogą wystąpić pewne problemy):

1. MUSI zwać się "RAD:".

2. Z pól pliku 'DEVS:MountList' (dla Workbencha 2.1 'DEVS:DOSDrivers/RAD') możesz zmieniać tylko pola: HighCyl, Buffers, Mount, DosType (pole to nie jest normalnie użyte, ale możesz je wprowadzić), natomiast nie wprowadzaj pól: FileSystem, Handler.

Mam nadzieję, że program ten przyda się każdemu posiadaczowi przełącznika Kickstartów.

Autorem programu jest Kamil Iskra.

2. PIC-IFF Converter 1.1.



PIC-IFF Converter służy do zamiany formatu zapisu rysunków z PIC na IFF. Pliki zapisywane są w foramcie IFF (HAM).

Format zapisu grafiki PIC: struktura zbioru

WORD Width - szerokość w pikselach (starszy bajt, młodszy bajt),

WORD Height - wysokość w pikselach,

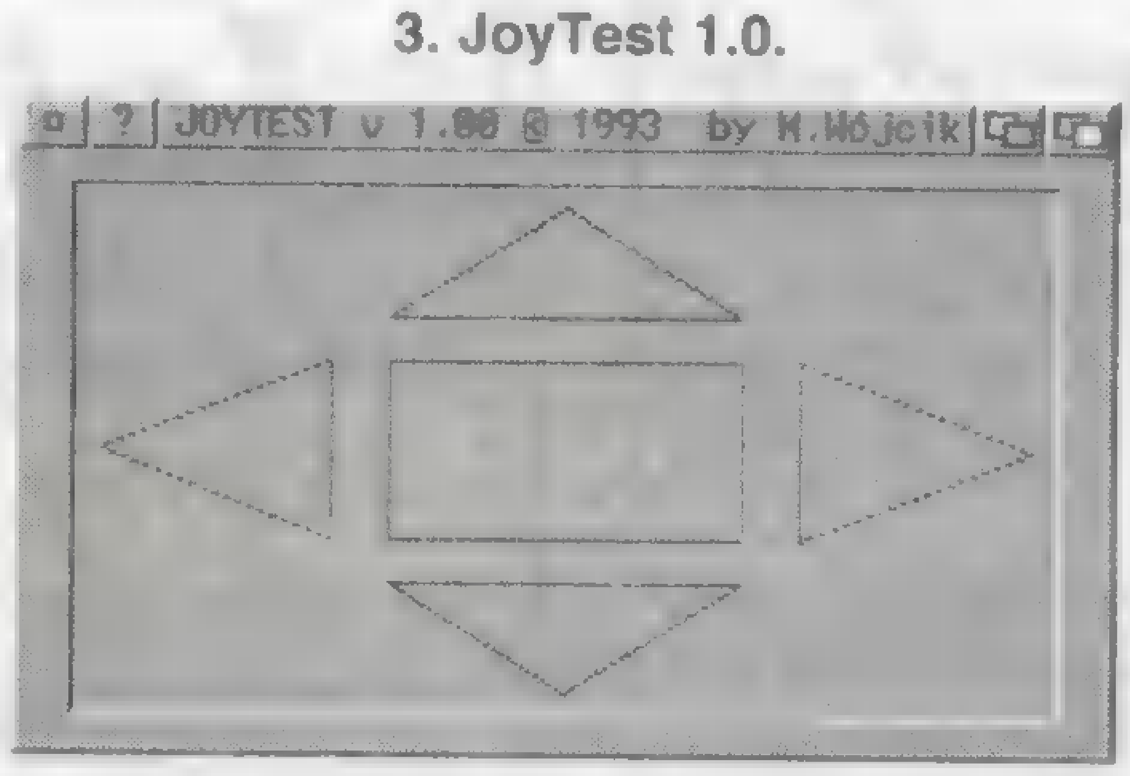
BYTE Red - ośmiobitowa wartość barwy składowej czerwonej dla piksela nr 1,

BYTE Green - ośmiobitowa wartość barwy składowej zielonej dla piksela nr 1,

BYTE Blue - ośmiobitowa wartość barwy składowej niebieskiej dla piksela nr 1.

Autorem programu jest Rafał Mantiuk.

3. JoyTest 1.0.



Program, jak sama nazwa wskazuje, służy do testowania joysticka (dla maniaków spolszczania: joystick, to manipulator drążkowy). Wielokrotnie zdarzyło mi się naprawiać uszkodzone joysticki, a nie mając pod ręką żadnej gry, nie miałem ich gdzie sprawdzić. W obecnej wersji program ten obsługuje tylko joysticki cyfrowe (czyli standardowe joysticki Amigi).

Program można uruchomić zarówno z Workbencha, jak i z CLI. Z CLI nie należy podawać żadnych parametrów. Program uruchomiony z CLI lub Shell utworzy własny proces, dzięki czemu CLI nie będzie blokowane - nie ma potrzeby używania komendy 'Run' czy 'RunBack'.

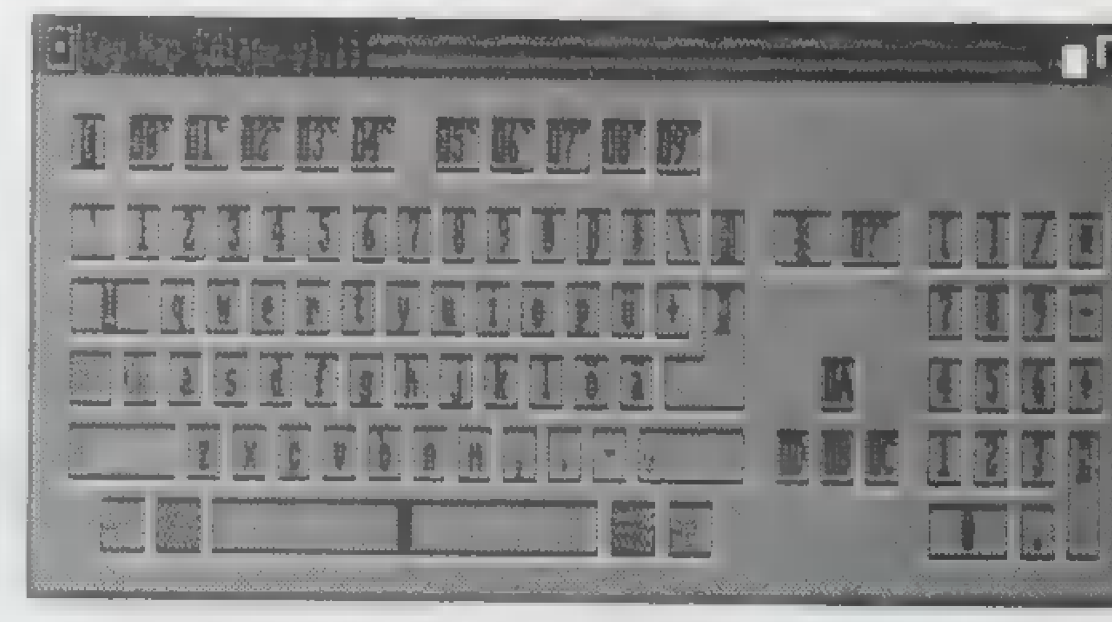
Z programu wychodzi się naciskając gadżet zamykania.

Na górnej listwie znajdują się standardowe gadżety oraz gadżet '?' służący do otwarcia okienka informacyjnego. Wychodzi się z niego naciskając gadżet 'Wyjście'.

W głównym oknie umieszczone są figury geometryczne przedstawiające poszczególne ruchy joysticka (oraz fire!). Każdy ruch jest sygnalizowany przez 'zapalenie się' danej figury geometrycznej.

Autorem programu jest Michał Wójcik.

4. KeyMapEditor.

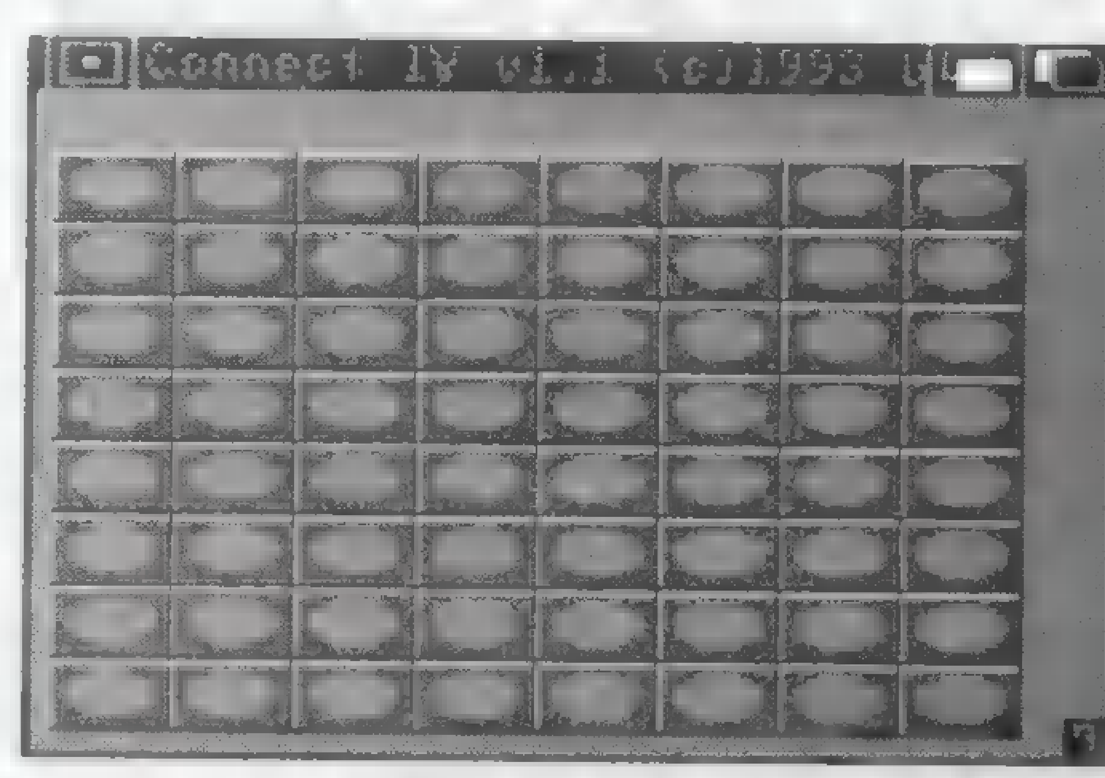


Jest użytecznym narzędziem pozwalającym na sprawną edycję mapy klawiatury standardowo dołaczanej do Amigowego Systemu Operacynego. Tak zmodyfikowany układ klawiatury jest gotowy do zainstalowania w systemie przy pomocy komendy SetMap.

Jak ogólnie wiadomo, prawie każdemu klawiszowi klawiatury komputerowej odpowiada jakiś jeden, ściśle określony, znak. Jednak w połączeniu z jednym lub kilkoma klawiszami specjalnymi istnieje możliwość uzyskania dodatkowych funkcji ([SHIFT], [ALT], [SHIFT]+[ALT], [CONTROL], [CONTROL]+[SHIFT], [CONTROL]+[ALT] i [ALT] +[SHIFT]+[CONTOL]). Tak więc prawie każdemu klawiszowi można w prosty sposób przypisać aż 8 różnych znaczeń. Właśnie KeyMapEditor pozwala na przypisanie klawiszom żądanych funkcji. Jest on alternatywą do systemowego KeYMapa, którego obsługa pozostawia wiele do życzenia.

Autorem programu jest Tim Friest.

5. Connect IV 1.1.



Connect IV jest grą logiczną przeznaczoną dla dwóch graczy (można grać z komputerem). Polega ona na połączeniu ze sobą czterech pól w jednakowym kolorze. Każdy z graczy dysponuje jednym kolorem. Ruchy wykonywane są na przemian. Aby wykonać ruch należy kliknąć lewym przyciskiem myszy na jednej z ośmiu kolumn (w przypadku ustawienia sterowania za pomocą klawiatury, należy nacisnąć klawisz od 1 do 8). Wygrywa ten,

kto pierwszy połączy w pionie, poziomie lub na którejś z przekątnych cztery swoje pola.

Opcje dostępne z menu:

Menu Project:
- New Game - wyczyszczenie planszy i rozpoczęcie nowej gry,
- About - informacja o autorze i programie,
- Help - krótka instrukcja obsługi,
- Quit - wyjście z programu.

Menu Oppenent:
- Player - Twoim przeciwnikiem jest drugi gracz,
- Computer - Twoim przeciwnikiem jest komputer.

Menu Control:
- Mouse - sterowanie za pomocą myszy,
- Mouse & Keyboard - sterowanie za pomocą myszy i klawiatury (klawisze od 1 do 8),
- p1-Mouse p2-Keybord - gracz pierwszy używa myszy, gracz drugi klawiatury.

Autorem gry jest Rafał Mantiuk.

6. Puzzle v1.1.

Puzzle jest to gra logiczna przeznaczona dla jednego gracza. Polega ona na ułożeniu poszczególnych elementów układanki w następującej kolejności:

1 2 3 4
5 6 7 8
9 A B C
D E F

Elementy można przestawiać za pomocą klawiszy kursora (klawisze ze strzałkami). Do ponownego przemieszania układanki służy gadżet z prawej strony okna. Liczba znajdująca się w dolnej części okna jest licznikiem Twoich ruchów.



Autorem gry jest Rafał Mantiuk.

7. Dobranoc - moduł muzyczny - świetny remix utworu pochodzącego z filmu dla dzieci (?!?).

8. Intro - krótki programik demonstracyjny napisany przez Rafała Mantiuka.

Autorem większości ikonek zamieszczanych na naszych dyskietkach jest Dariusz Zwierzyński.

Prosimy o uwagi o naszej bibliotece. Piszcie do nas jakich programów poszukujecie i jakie powinny się w niej znaleźć. Czekamy również na owoce Waszej pracy.

Jak zamawiać i ile to kosztuje?

Dyskietki PD można zamawiać przesyłając pieniądze na nasze konto przy pomocy blankietu z Amigowca, podając symbol dysku PD (#1 do #12, #1-#6/93, #1/94 oraz "A", "B", "C") i oczywiście swój czytelny adres. Nasza dyskietka kosztuje 35 tysięcy złotych (wliczone są koszty dyskietki, przesyłki, nalepki, opakowania, itp.). W przypadku przesyłki za zaliczeniem pocztowym zamawiający pokrywa koszty zaliczenia. ❑



Dyski muzyczne

Marcin Gackowski

Muzyka jest prawie dla każdego czymś bardzo ważnym. Słuchamy jej wszyscy. Z radia, telewizji, kaset, płyt i kompaktów. Jeżeli chodzi o mnie, często słucham jej także z mojej Amigi. Ciągle udoskonalane muzycznie dema i dyski muzyczne sprawiają, że włączam czasami komputer tylko po to, aby posłuchać muzyki. Coraz lepszej i lepszej...

Niniejszy odcinek World Scene poświęcony został w całości muzyce, a właściwie dyskom muzycznym, które pojawiły się w ostatnim czasie na scenie.

Po przesłuchaniu zachodnich i krajowych dem z ostatnich miesięcy, wydawać by się mogło, że amigowa muzyka sprowadziła się wyłącznie do techno. Jednak w opisywanych tutaj produkcjach ani jeden utwór nie ma nic wspólnego z tym gatunkiem muzyki. Widać więc, że czołowi muzycy nie zawężają swojej wyobraźni muzycznej do rytmicznej łupaniny, a eksperymentują i tworzą naprawdę ciekawe kompozycje. Pozostaje tylko mieć nadzieję, że moda na techno przeminie i nie dojdzie do tego, że słowo Amiga będzie synonimem wyrazu techno.

A teraz krótko o tym, co miłego dla ucha pojawiło się ostatnio na scenie:

Book of Songs / Complex

Dwudyskowa produkcja, która szokuje jakością sampli i staranną oprawą graficzną. Każdemu utworowi towarzyszy dopasowana tematycznie ilustracja oraz stosowny tekst. Kompozycje są rewelacyjne. Szczególnie szybki, jazzująco-funkujący Blowaway, nastrojowy Touch of Green oraz Real Motivation z realistycznym saksofonem. Muzycy grupy Complex: Jugi i Delorean poświęcili zapewne wiele czasu na tak ogromne dopracowanie utworów. Efekt słyszalny jest od pierwszych taktów każdej kompozycji. Book of Songs to ogromny krok do przodu w dziedzinie amigowej muzyki. Dla mnie - rewelacja....

Sound Barrier / Balance

Również dwudyskowy, jednak trochę gorszy musicdisk. Gorszy tylko pod względem muzycznym, bowiem oprawa graficzna i design są bardzo dopracowane. W tego rodzaju produkcjach muzyka jest rzeczą najważniejszą i na nią chciałbym zwrócić uwagę. Utwory nie są złe. Są miłe dla ucha, melodyjne i dopracowane. Razi jednak brak oryginalności i pewne zawężenie tematyczne. Mimo tego, kompozycje takie jak Bionic Savage, Hallucination, Bjon Wold i przede wszystkim Unspoken Love zostają na długo w pamięci. Autorami muzyki w Sound Barrier są Blue Fox i Subject.

Imperial Tunes 2 / Parasite

Jeden dysk. Tylko dwa utwory. A mimo to wysoka pozycja w notowaniach zachodnich magazynów. Chyba nie trudno zgadnąć, że ma to związek ze zwalającą z nóg muzyką. I tak rzeczywiście jest. Pierwszy utwór, autorstwa Liquid'a, to cudowny rockowy kawałek z ciekawie wplataną harmonijką i niesamowitą gitarą. Drugi - Judgement Day,



skomponowany przez Slide'a jest wzniosłą, nastrojową kompozycją, zahaczającą często o klimaty psychodeliczne. Jednym słowem - odlot.

Kugle P0len / Static Bytes



Zajmuje jeden dysk i jest kolejnym rewelacyjnym produktem, który pokazuje na ile jeszcze stać poczciwą Amigę w dziedzinie muzyki, jeżeli zasiądzie do niej odpowiednia osoba. Znakomite pomysły, jak chociażby ten ze śpiewem w Magic Universe. Wspaniały rytm w Blush Funk. Smokybar przypominający melodie z "Żądła". Spokojny Sad Dream, czy wreszcie Syntetic Quartet, czyli trochę muzyki poważnej. Zróżnicowanie stylistyczne, czysto brzmiące sample, niezła technika i pomysły. To wszystko razem pozwala mniemać, że bliżej nieznany muzyk - Cutcreator już niedługo będzie gościł w czołówce światowej. Kugle P0len polecam wszystkim. Mam pewność, że spodoba się każdemu.

Pastel Vanilla / Nova

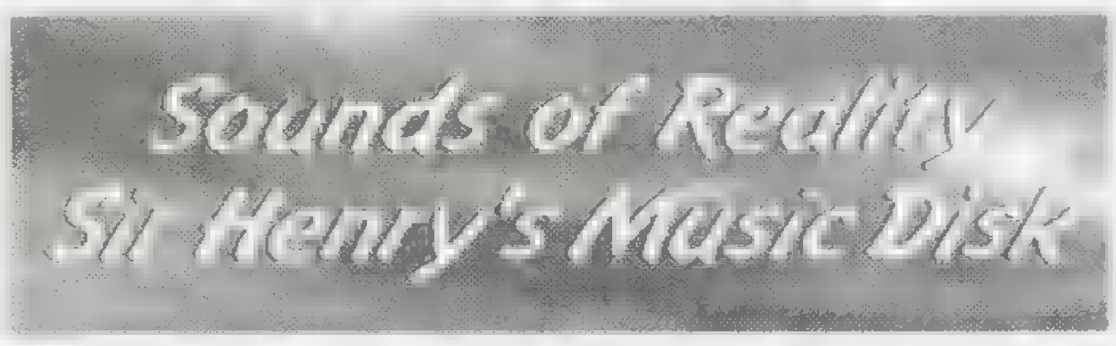


Zdecydowanie najgorszy z opisywanych tutaj produktów. Na dwóch dyskach umieszczono sporo kompozycji, lecz do przesłuchania wszystkich w całości potrzeba dużo silnej woli. Muzyka jest monotonna, niezbyt przemyślana i chwilami nużąca. Technicznie nie można kompozytorowi nic zarzucić. Blue Silence, gdyż to on jest autorem wszystkich utworów, powinien (według mnie) ograniczyć się do jedne-



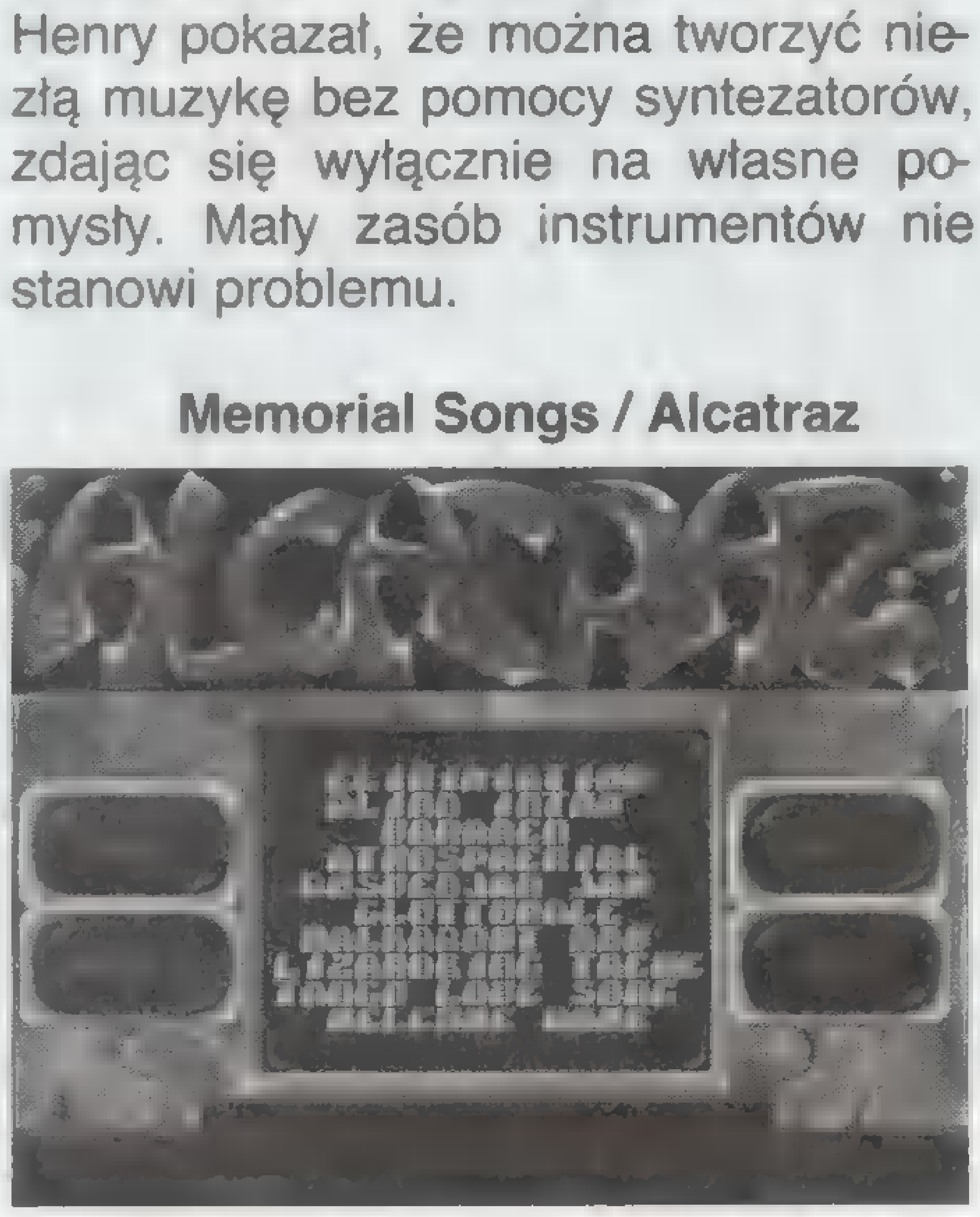
go dysku. Wtedy produkt byłby bardziej strawny. Najlepsze kompozycje to Aven i Try. Inne (najczęściej przeróżne Funk'i) są dosyć jednolite, więc produkt ten polecam jedynie kolekcjonerom.

Sounds of Reality



Na pierwszy rzut oka nie jest to dzieło rewelacyjne. Statyczny ekran bez scroll'a, jeden logos i opis klawiszy. Żadnych bajerów. A jednak, gdy wsłuchamy się w to co płynie z głośników, od razu zmieniamy zdanie o tym produkcie. Sześć kompozycji zróżnicowanych stylistycznie od disco (Digital Dancefloor), poprzez blues (Mouselephant Blues), balladę, nowoczesne pop-regge (Morning Love) do akordeonowych wariacji (Smuggler). Mimo, tego, że sample nie są rewelacyjne, a technika kompozytora nienajlepsza, całość brzmi naprawdę świetnie. Sir Henry pokazał, że można tworzyć niezłą muzykę bez pomocy syntezatorów, zdając się wyłącznie na własne pomysły. Mały zasób instrumentów nie stanowi problemu.

Memorial Songs / Alcatraz



Najnowszy z opisywanych musicdisków i wydaje się, że najlepszy. Znany nie tylko amigowcom Lizardking prezentuje w tym produkcie 10 swoich utworów. Żadnemu nie można nic zarzucić, wręcz przeciwnie, można tylko podziwiać i chwalić za melodyjność, idealną harmonię, pomysłowość i wspaniale brzmiące sample. Najlepsze utwory? Na pewno Tango Love Song, Lizardking Theme, Elimination, Welcome Home i Blind Intro z interesującym głosem z voicecodera. Memorial Songs jest kolejnym dyskiem muzycznym, który potwierdza tezę, że nie wygląd i design jest najwżniejszy w tego typu produkcjach. Owszem, czynniki te decydują w ogromnej mierze o wartości typowych dem, jednak musicdiski zostały wymyślone do słuchania muzyki i ona jest tu najważniejsza.



To wszystko w tym odcinku. Na jesień planowane są liczne party (w tym Polish Autumn C-party w Poznaniu), czekajmy więc cierpliwie na nowe dema. W następnym odcinku World Scene chciałbym powrócić do tradycyjnej już formy tej rubryki, czyli prezentacji rasowych dem, na które warto zwrócić uwagę. ❑



LISTA DEMONÓW

Artur Łukasik, Sebastian Klomski

Witamy po raz kolejny! Tym razem niewielkie zmiany na naszej liście. Dotyczy to zarówno listy zagranicznej, jak i krajowej. Stan ten zmieni się zapewne po otrzymaniu przez nas wyników z PARTY III w Danii.

ZAGRANICA

GRUPY:

1. Sanity
2. Kefrens
3. Lemon
4. Melon Dezign
5. Andromeda
6. Silents
7. Complex
8. Dual Crew - Shining
9. Spaceballs
10. Movement

Grupa Melon spada na coraz niższe pozycje, zaś do góry pnie się bardzo aktywna grupa Lemon. Grupa ta składa się w większości z członków grupy Anarchy (uważanej przez wielu ludzi za najlepszą na świecie). Formację Lemon podobnie jak jej poprzednika - Anarchy cechuje niezwykła produktywność. Wydawałoby się, że grupa Complex zniknie ze sceny, lecz na party 680xx Convention organizowanym przez Complex, Sanity, Supplex, White Status wydała demo pod tytułem PARADIGMA. Dzięki temu Complex ponownie wskoczyła na topy.

DEMA:

1. *Hardwired* /Cronics & The Silents
2. *Desert Dream*/Kefrens
3. *Extension*/Pygmy Projects
4. *Rink a Pink*/Lemon
5. *How to Skin a Cat* /Melon Dezign
6. *Groovy*/Lemon
7. *D.O.S.*/Andromeda
8. *State of The Art*/Spaceballs
9. *World of Commodore*/Sanity
10. *Guardian Dragon II*/Kefrens

Z nowych dem jedynym godnym uwagi było Rink a Pink wyżej wspomnianej grupy Lemon.

DEMA PLIKOWE:

1. *Interference*/Sanity
2. *Mindriot*/Andromeda
3. *3D Demo II*/Anarchy
4. *Paradigma*/Complex
5. *Announce*/Lemon
6. *Transpiersky*/The Silents
7. *Booo*/Melon Dezign
8. *Terminal Fuckup*/Sanity
9. *OptimumFuckup*/Sanity
10. Joyride/Phenomena

Objawieniem ostatnich dni jest interesujące dentro (czyli coś pośredniego pomiędzy intrem, a demem - chociaż w dzisiejszych czasach różnice te często zacierają się) Paradigma stworzone przez powracający Complex.

INTRA 40 KB:

1. *Tetris*/Melon Dezign
2. *Shinig*
3. *Lemon*
4. *Virtual Dreams*
5. *Arrilfoll*/Dual Crew & Shinig
6. *Stellar*
7. *No Pain No Gain*/TRSI
8. *The Silents*
9. *TEK 10.Spaceballs*

MUSIC DISKI:

1. *Mirror*/Andromeda
2. *Memorial Songs*/Alcatraz
3. *Jesterday*/Sanity
4. *Chromagic* /Dual Crew & Shining
5. *Crystal Symphonies II* /Phenomena
6. *Kuglep0len*/Static Bytes
7. *Book of Song*/Complex
8. *Turmoil*/Sanity
9. *Music Dream ii*/Phenomena
10. *Crystal Symphonies I* /Phenomena

Na czwartej pozycji nowy dysk muzyczny pracowitej w ostatnim okresie grupy Dual Crew & Shining.

KODERZY:

1. *Chaos*/Sanity
2. *Hannibal*/Lemon
3. *The Spy*/Crionics
4. *Laxity*/Kefrens
5. *Lone Starr*/Spaceballs
6. *Slammer*/Melon Dezign
7. *Bannasoft*/Melon Dezign
8. *Microforce*/Sanity
9. *Dr.Jekyll & Mr.Hyde*/Andromeda
10. *Performer*/Melon Dezign

GRAFICY:

1. *Facet*/Lemon
2. *R.W.O.*/Kefrens
3. *Cougar*/Sanity
4. *Peachy*/TRSI
5. *Mack*/Melon Dezign
6. *Fairfax*/Andromeda
7. *Rack*/Absolute
8. *UNO*/Scoopex
9. *Walt*/Melon Dezign
10. *Havok*/Sanity

Na liście pojawił się Rack z węgierskiej grupy Absolute. Jeszcze nie tak

dawno tworzył znakomite prace pod szyldem grupy Majic 12. Jego najlepsze grafiki mogliśmy podziwiać w niesamowicie dopracowanym graficznie demie pt. Ray of Hope 2.

MUZYCY:

1. ***Jester*/Sanity**
2. ***Audiomonster*/Melon Dezign**
3. ***Lizard King*/Alcatraz**
4. ***Laxity*/Kefrens**
5. ***Nuke*/Lemon**
6. ***Romeo Knight*/TRSI**
7. ***Tip & Mantronix*/Phenomena**
8. ***Bit Arts*/Complex**
9. ***Jesper Kyd*/The Silents**
10. ***Virgill*/Masque**

POLSKA

GRUPY:

1. **Old Bulls (*)**
2. **Deform**
3. **Applause**
4. **Union**
5. **Alchemy**
6. **W.F.M.H.**
7. **Mad Elks**
8. **Suspect**
9. **Joker**
10. **Damage**

Po ostatnim party, które odbyło się w Poznaniu (sprawozdanie z niego można przeczytać w numerze 9-10/1993 Amigowca) krążyło wiele plotek wokół grupy Old Bulls. Przedstawiamy poniżej skład tej grupy:

Arcus - swapowanie
Kopyto - swapowanie
Processor - kodowanie
Yahoo - kodowanie

oraz następujące podgrupy:

Gel Dezign:

Accord - muzyka
Zefir - grafika

TILT:

Bukka - kodowanie
Hudi - kodowanie
Sabe - grafika, swapowanie, kodowanie

V.I.P.'s:

Alwen - kodowanie,swapowanie
Blind - grafika
Walker - swapowanie

Grupa Old Bulls jak przystało na najproduktywniejszą grupę na polskiej scenie szykuje w najbliższym czasie następujące produkcje:

Raytraced Dreams III - slide show (AGA) - TILT - kwiecień 94
Imagination - trackmo (AGA) - TILT - sierpień 94
Grey Dreams - slide show - V.I.P.'s - kwiecień 94
Cure - dysk muzyczny - Gel Dezign - kwiecień 94
Autobiography II - Old Bulls (całe) - luty 94
40k intro - Old Bulls - kwiecień 94
Smile - dentro - TILT - kwiecień 94
Nightmare - trackmo (AGA) - TILT - grudzień

DEMA:

1. ***Impulse II/TILT*/Old Bulls**
2. ***Deformations*/Deform**
3. ***Fugazi*/Old Bulls**
4. ***Technological Death*/Mad Elks**
5. ***Hallucinations & Dreams*/Union**
6. ***Software*/W.F.M.H.**
7. ***Marchewki*/Alchemy**
8. ***PRO.s.i.a.k*/FCI**
9. ***The Return*/Joker**
10. ***Uquala*/Damage**

DEMA PLIKOWE:

1. ***Party-zanci*/Joker**
2. ***Bara Bara*/Applause**
3. ***Impulse-preview*/Old Bulls (TILT)**
4. ***No Name*/Suspect**
5. ***Varathron*/Suspect**
6. ***Autobiography*/Old Bulls**
7. ***Anam*/Investation**
8. ***Xenium*/Old Bulls**
9. ***Defene coś tam*/Luzers**
10. ***Ingis*/Investation**

DYSKI MUZYCZNE:

1. ***Vengeance*/Beta Team**
2. ***Orgasm*/Old Bulls (TILT)**
3. ***Waiting for Another Beer*/Joker**
4. ***Music Tracks*/Suspect**
5. ***Kadi Sound Disk III*/Joker**
6. ***Hunt for the red kret*/FCI**
7. ***Illusion*/Old Bulls**
8. ***Techno Fuse*/Investation**
9. ***Kadi Sound Disk II*/Joker**
10. ***Sigh*/Old Bulls**

KODERZY:

1. ***Musashi*/Union**
2. ***Hudi/TILT*/Old Bulls**
3. ***Bukka/TILT*/Old Bulls**
4. ***Vico*/Alchemy**
5. ***Robin*/W.F.M.H.**
6. ***Kane*/Suspect**
7. ***Tom*/Union**
8. ***Dak*/Mad Elks**
9. ***GBH*/Joker**
10. ***Miklesz*/Damage**

MUZYCY:

1. ***XTD*/Union**
2. ***Peters*/Joker**
3. ***Pic Saint Loup***
4. ***Accord/Gel Dezign*/Old Bulls**
5. ***Mr.Root*/Union**
6. ***Dreamer*/FCI**
7. ***Snoopy*/Union**
8. ***Seq*/Union**
9. ***Kadi*/Joker**
10. ***Passat*/Funzine**

GRAFICY:

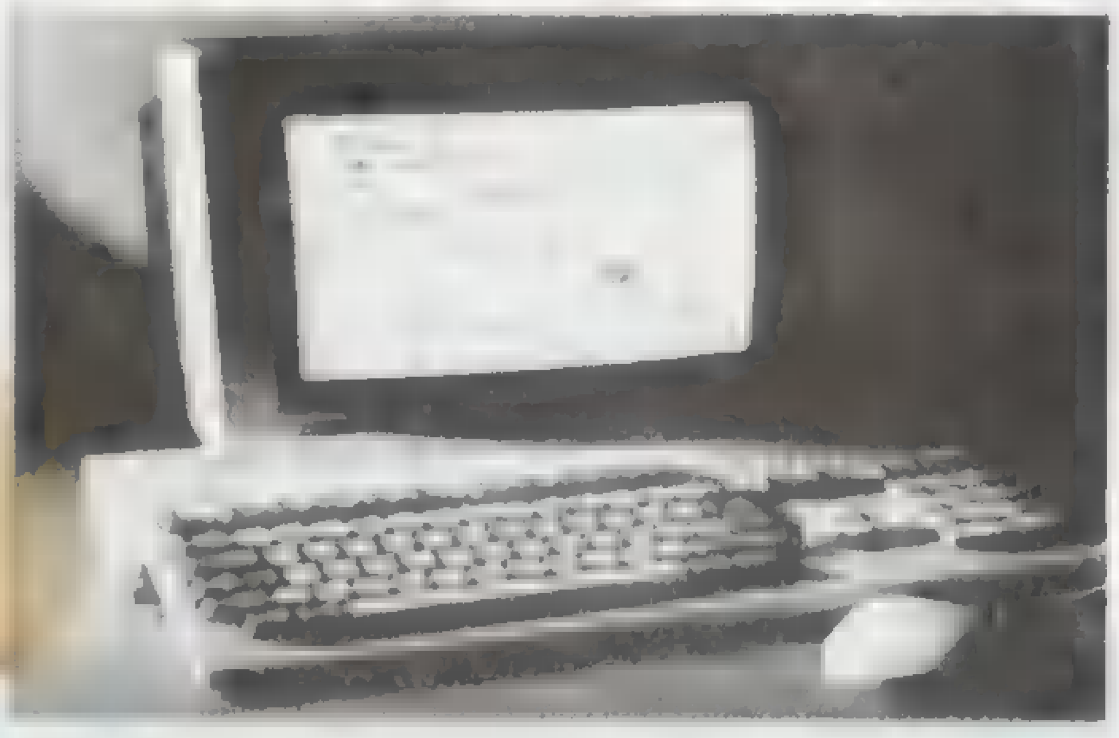
1. ***TPP*/Applause**
2. ***Animal*/Union**
3. ***Sabe/TILT*/Old Bulls**
4. ***Seq*/Union**
5. ***Pluton*/Union**
6. ***Python*/Union**
7. ***Frodo*/Alchemy**
8. ***Zefir/Gel Dezign*/Old Bulls**
9. ***Rys*/Joker**
10. ***Berserker*/Investation**

MAGAZYNY DYSKOWE:

1. ***Zig Zag*/Union**
2. ***Zasmashka*/Alchemy**
3. ***Thing*/Applause**
4. ***Paper White*/Luzers**
5. ***Fat Agnus*/Investation**
6. ***Imazine*/Saint Group**
7. ***X-mag*/Union**
8. ***Krawężnik*/Status OK**
9. ***Nie z tej beczki*/Lamers**
10. ***Poczytaj Mi Mamo***

Wszystkie wiadomości pochodzą z pewnych źródeł i są w 100 procentach sprawdzone.

Głosujemy na dowolną ilość pozycji z każdego działu, przyznając im od 1 do 10 punktów. Głosy, propozycje, swoje prace, uwagi, a także najnowsze programy demonstracyjne i magazyny dyskowe prosimy nadsyłać na adres redakcji. ❑

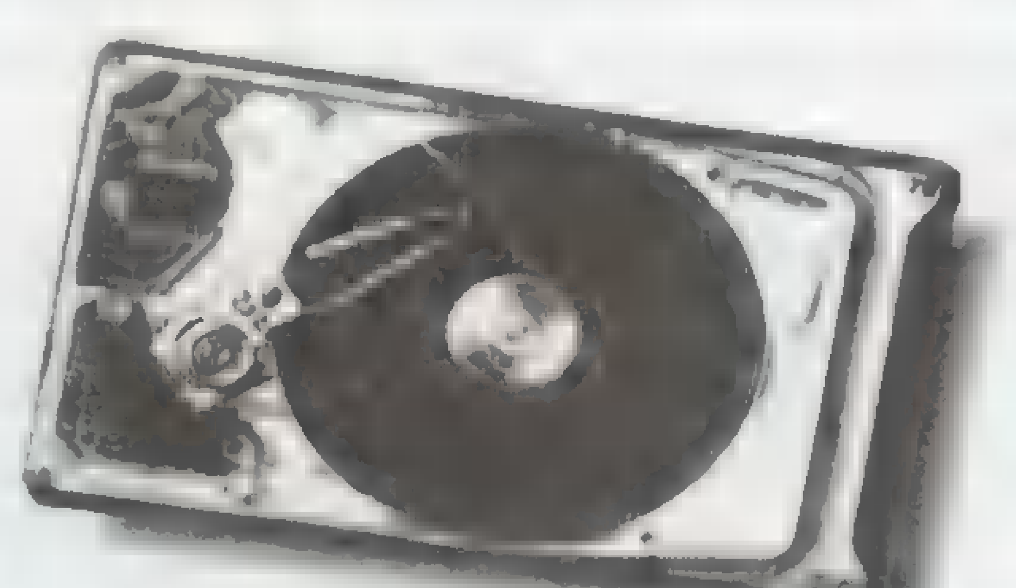


producent: Copperhead Technologies
dystrybutor:
Copperhead Technologies,
104 Jay Street,
Schenectady,
NY 12305, USA,
tel. 010 1 518 346 3894
zgodność: A500
cena: około 225$

Z małej Amigi - duża, czy mniejsza Amiga z A500 A1200!

Swego czasu, w jednym z czarno-białych wydań AMIGOWCA opisywaliśmy możliwość przestosowania AMIGI 500 na Amigę prawie 2000. Wystarczyło dokupić zestaw o dźwięcznej nazwie "Bodega Bay" i już mieliśmy dwie Amigi zamiast jednej. Tym razem można bez większych problemów przerobić naszą ulubioną A500 na jeszcze bardziej ukochaną A1200. I to wszystko za pół ceny. Wydaje się to niewiarygodne, tym bardziej, że do redakcji nie dotarły żadne zajęcia pokazujące taką Amigę 500-1200. Amerykańska firma Copperhead Technology specjalizująca się w handlu używanym sprzętem dostarcza na zamówienie zestaw, który przekształci Waszą A500 w Amigę 1200. To co zostanie przekształcone to oczwyiście układy graficzne - ■ zwykłych zamieniają się w AGA kompatybilne (w systemie PAL). Po- ■ tym wymieniany jest kickstart na 3.0, pamięć graficzna Chip RAM jest rozszerzana do 2MB oraz dokładany jest kontroler IDE umożliwiający podłączenie twardych dysków typu AT. Specjalny adaptor umożliwia podłączenie starej, dobrej klawiatury ■ A500 do nowej obudowy. Również w podobny sposób można wykorzystać wewnętrzną stację dysków i zasilacz. Producent głosi, że nawet mało doświadczony użytkownik jest w stanie dokonać takiej przeróbki. Oczywiście są przy tym niezbędne pewne modyfikacje naszej staruszki - należy wyciąć plastikową kolumnę, która oddziela porty z tyłu A500 i mały kawałek plastiku, który styka górę i dół obudowy w środku komputera.

Gdy już zainstalujemy cały ten zestaw to nasz "nowy" komputer działa tak jak normalna Amiga 1200. Ma nawet charakterystyczne dla niej wyjście PCMCIA i klapkę, przez którą można wyprowadzać na zewnątrz np. porty kart graficznych.

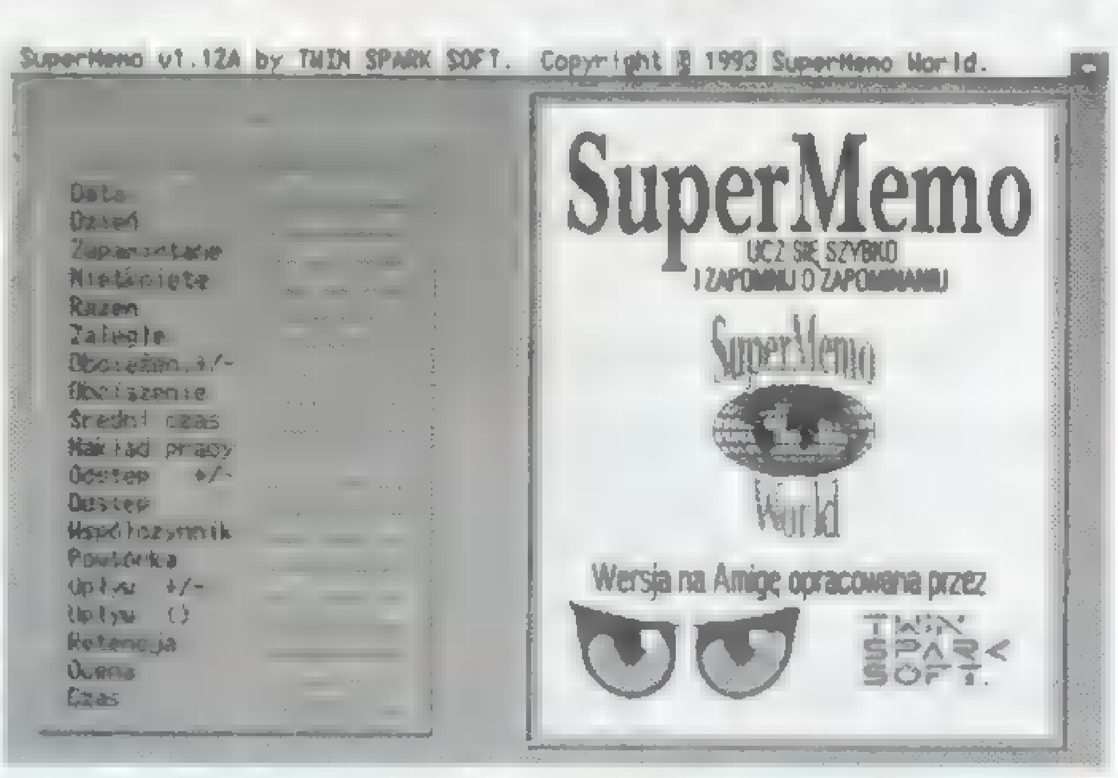


producent: MAXTOR
dystrybutor:
MAXTOR Europe Ltd,
Unit 3 Woking 8,
Forsythe Road, Woking, Surrey,
GU 21 5SB
zgodność: A600, A1200 (2.05, 3.0)
cena: ?

Twardziel do PCIMCI – śpiewką przyszłości?

Jak zapewne zdążyli się już zorientować posiadacze Amig 600 i 1200, ich przyjaciółki są wyposażone w dodatkowy dziwny otwór z boku, do którego nie dostarczono zatyczki. Jest to otwór portu PCMCIA. Jego wykorzystanie do tej pory opierało się głównie na zasobnym portfelu. Każdy kogo było na to stać mógł zakupić sobie rozszerzenie pamięci, które potem mogło mu służyć do szybkiego przenoszenia danych pomiędzy dwoma komputerami.

Ostatnio powstał miniaturowy twardy dysk, który pasuje do PCIMCI. Producentem tego cuda jest firma MAXTOR, znana ze swoich większych produktów. Dysk jest wielkości karty kredytowej (4.5 cm) i ma pojemność 105 MB i jak głosi producent jest to największa pojemność jaką osiągnięto na tego typu dysku. Montaż polega na zręcznym wepchnięciu do slotu PCIMCI. Dysk ten ma służyć jako doskonały przenośnik danych pomiędzy komputerami. Już widzę zazdrosne miny redakcyjnych kolegów, którzy o czymś takim właśnie marzą. Przenoszenie danych mogłoby się odbywać pomiędzy wszystkimi komputerami wyposażonymi w sloty PCMCIA, ■ więc pomiędzy np. Amigami 600, 1200, laptopami i innymi komputerami typu notebook. Na dysku tej pojemności zmieści się praktycznie wszystko. Krótko mówiąc można bez większych problemów nosić ze sobą całe biuro. Standard PCMCIA zapewni także dwukrotnie większą szybkość przesyłania danych. Niesamowite jest również zabezpieczenie tego dysku przed wszelkiego rodzaju urazami. To nic, że uda Wam się spaść ze schodów i skręcić kark - Wasz komputer, będzie działał dalej, he, he. Jaka będzie cena tego cacka jeszcze nie wiadomo. Nie wiadomo również kiedy pojawi się ono na rynku.



producent: TWIN SPARK SOFT
dystrybutor:
TWIN SPARK SOFT,
31-939 Kraków,
os. Kolorowe 9/16,
tel. 44-43-68,
fax. 47-72-99
zgodność: 1.x, 2.x, 3.x
cena: 390 tysięcy

SuperMemo - program dla Twojej pamięci

SuperMemo jest programem jednocześnie bardzo prostym w obsłudzie i posiadającym niesamowite wręcz możliwości. Zasada działania programu jest prościutka:

Jeśli chcesz się uczyć, to my Ci w tym pomożemy!

Przy pomocy SuperMemo można się uczyć praktycznie wszystkiego. Poczynając od tak klasycznych rzeczy jak czasowniki nieregularne, poprzez słownictwo (prawie) dowolnego języka na znakach drogowych wcale nie kończąc. Oprócz znaków drogowych i rozwiązywania zadań przygotowujących do egzaminu na prawo jazdy, SuperMemo pozwala również na naukę rozpoznawania np. obrazów największych mistrzów.

Sam program jest niewielki i łatwo uruchamia się z dyskietki. Można oczywiście zainstalować go na dysku twardym. Obsługa odbywa się w języku polskim, ■ do programu dołączono bezpłatnie polskie drivery do niektórych drukarek autorstwa Ks. Jana Pikula.

Użytkowanie programu nie powinno przysporzyć nikomu problemów. Wraz z dyskietką otrzymujemy książeczkę - instrukcję, która wyjaśnia wszystkie szczegóły.

SuperMemo jest amigowską wersją programu działającego do tej pory tylko na IBO-Lach i jako taki, zaliczył już sporo nagród, a w tym: dotarł do finału konkursu SOFTWARE FOR EUROPE CeBIT'92 oraz zdobył pierwszą nagrodę na targach SOFTARG'92.

Program pomaga w nauce, a ściślej w zapamiętywaniu dowolnej ilości informacji. Jego zaletą jest dostosowanie się do możliwości i sposobu uczenia się użytkownika oraz możliwość tworzenia własnych baz danych. Jest on praktycznym wykorzystaniem najnowszych badań dotyczących uczenia się i zapominania, w taki sposób, aby można było zapomnieć o tym drugim.

CYBERPunk

01010101101011001

Tomasz Łobada

Często zdarza się, że nadanie nowej nazwy starym rzeczom pozwala odkryć je na nowo. Nie inaczej było z tym, co obecnie zaczęto nazywać Cyberpunkiem.

Skąd to się właściwie wzięło?

Dla pokolenia wychowanego w latach osiemdziesiątych takie rzeczy jak: telewizja satelitarna, komputery, konsole do gier, video czy odtwarzacze CD stanowią naturalne elementy świata. To właśnie z nich wyrośli hackersi, swobodne ptaki komputerowych sieci. Ich to działania stworzyły podstawę nowego ruchu, a filozoficznej otoczki dostarczyła niezawodna literatura SF. Dzięki staraniom miesięcznika "Fenix" polscy czytelnicy mogli zapoznać się z twórczością najbardziej znanych autorów nowego kierunku literatury fantastycznej: Cyberpunku. Na fali rosnącej popularności pojawiła się na księgarskich półkach, kultowa już książka Williama Gibsona: "Neuromancer" (warto wspomnieć, że oryginalną okładkę tej powieści wykorzystało wydawnictwo Amber w książce Philipa Dicka "Trzy stygmaty..."). Powieść ta usystematyzowała i dostarczyła niezbędnego dla młodego ruchu nazewnictwa.

O nazewnictwie słów kilka.

Co kryje się pod nazwą Cyberpunk? Nie trudno się domyślić, że wyraz ten powstał z połączenia dwóch słów: CYBER (skrót od cybernetyczny) i PUNK. Słowo "cybernetyczny" pochodzi z języka starogreckiego i znaczy "kierujący". Współcześnie jest to dziedzina nauk ścisłych, zajmująca się "układami o znacznym stopniu samosterowności", z kolei pod tym pojęciem kryją się "układy o wysokim stopniu komplikacji", czyli państwa, miasta, ludzie, komputery i sieci komputerowe. Dla nas najbardziej interesujące są dwie ostatnie pozycje sugerujące, że "CYBER" będzie oznaczał: świat komputerów i sieci komputerowych. Druga część nazwy: PUNK - jest nieodłącznie kojarzona z ruchem młodzieżowym, wyrosłym na tle muzyki nowofalowej lat siedemdziesiątych i osiemdziesiątych. W szerszym znaczeniu oznacza grupę ludzi, która ze względu na swoje zainteresowania i ideologię chce wyróżnić się z otaczającego ją świata.

Drugim niezwykle ważnym i nierozerwalnie związanym z Cyberpunkiem pojęciem jest: "Cyberprzestrzeń" ("Cyberspace") - czyli tysiące kilometrów przewodów, światłowodów, setki komputerów, baz danych, elektronicznych skrytek, słowem cały wspomniany wcześniej świat sieci komputerowych.

Trzecia nazwa związana jest z ostatnim krzykiem komputerowej techniki. To "rzeczywistość wirtualna" ("Virtual Reality"). Komputerowy ekran został w niej zastąpiony specjalnymi stereoskopowymi "goglami". Ten hełm XXI-wiecznego rycerza zaopatrzony został w ciekłokrystaliczne wyświetlacze, sterowane super szybkimi komputerami graficznymi. Można teraz nie tylko patrzeć na komputerowy świat, ale również wejść do jego wnętrza i rozejrzeć się (specjalne czujniki przekazują ruch głowy do komputera, a ten z kolei odpowiednio przemieszcza obraz widziany w okularach).

Co się za tym kryje?

Każda epoka ma swój własny wariant "zbuntowanej młodzieży". Koniec lat sześćdziesiątych przyniósł ruch hippisowski, natomiast lata siedemdziesiąte "nową falę". Później nastąpił długotrwały bezruch. Dopiero teraz otworzyła się możliwość zapoczątkowania czegoś nowego. Miejsce fanatycznych wyznawców określonego gatunku muzyki zaczynają zajmować nie mniej zatwardziali zwolennicy "jedynie słusznych i najlepszych" komputerów. Dla coraz szerszego grona nastolatków informatyka jest czymś więcej niż zabawą, to sposób na życie. Ludzie tacy, zaczynają odczuwać potrzebę zamanifestowania swojej odrębności w stosunku do innych "zwykłych śmiertelników", poprzez specyficzną gwarę, ubiór lub zachowanie. To właśnie zalążki Cyberpunku.

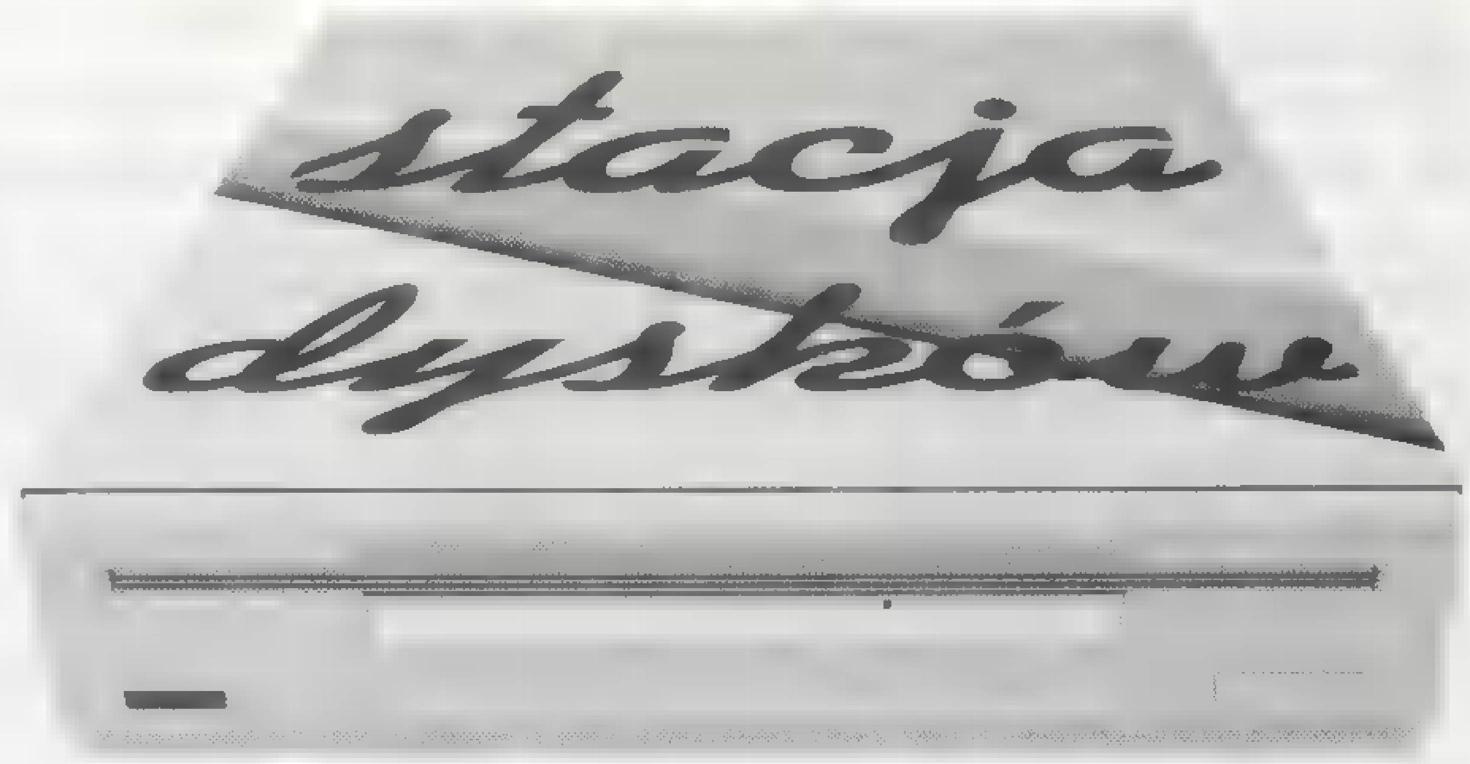
Jak to wygląda w praktyce?

Cóż, na razie Cyberpunkowcy mają jedynie możliwość korzystania ze specjalnych "skrzynek kontaktowych". Skrzynka taka to jeden lub kilka komputerów podłączonych do sieci, z których można korzystać za pomocą modemu podłączonego do naszego domowego komputera. Idea "skrzynek" polega na tym, że w wyznaczonym dla siebie miejscu pozostawia się wiadomości (najczęściej w postaci pliku tekstowego), którymi chcemy podzielić się z innymi użytkownikami sieci. Mogą to być zupełnie osobiste wyznania, najnowsze wiadomości z komputerowego świata lub cokolwiek innego. Takich miejsc wymiany danych jest już sporo i ciągle przybywa. Pozwala to na szybką, zupełnie nową jakościowo, wymianę informacji.

A co dalej?

Szybki rozwój elektroniki przyzwyczaił nas do ciągłych zmian i niespodzianek. To od nich w głównej mierze będzie zależał kształt przyszłego ruchu cyberpunkowego. Każdy kto obejrzał znakomity film pt. "Kosiarz umysłów", powinien mieć już niejakie pojęcie o przyszłości sieci komputerowych. Otwarte systemy z wizualizacją wszystkich operacji możliwych za sprawą "bezpośredniego wejścia" w komputerowy świat nie są dziś już tylko domeną literatury SF. Kto wie? Może przyszłość należeć będzie do bladych, niewyspanych chudzielców o rozmarzonych oczach, dla których "rzeczywistość wirtualna" będzie nie mniej realna od tej, widzianej na co dzień za oknem. ❑

AMOS i stacja dysków



Tomasz Hrycyniak

Tym razem chciałbym przedstawić zasady współpracy programów AMOSA ze stacją dysków. Jest to zagadnienie interesujące przede wszystkim osoby piszące programy użytkowe. Także gry często zawierają opcje zgrywania ich stanu na dyskietkę. Myślę więc, że temat ten zainteresuje wszystkich pragnących pisać własne programy w AMOSIE.

Starym zwyczajem rozpocznę od opisania komend podstawowych. W tym wypadku będą to instrukcje kontrolne przeznaczone do odczytywania katalogu, sprawdzania obecności plików na dysku, tworzenia katalogów, itp. Zatem zaczynamy:

Dir ścieżka$/w
ścieżka$ - Ścieżka dostępu do katalogu.
w - Opcja powodująca wyświetlenie katalogu w dwóch kolumnach.
Komenda ta powoduje wyświetlenie na ekranie spisu plików danego katalogu. Opcja "w" powoduje ułożenie spisu w dwóch kolumnach. W przypadku pominięcia definicji ścieżki, wyświetlony zostanie spis katalogu bieżącego.

Dir$
Jest to funkcja przyjmująca wartość określającą aktualny katalog dysku. Wykorzystać można ją np. tak:
Print Dir$
Spowoduje to wypisanie na ekranie scieżki dostępu katalogu bieżącego.
Dir$="Df0:libs/"
Takie użycie funkcji "Dir$" powoduje przyporządkowanie jej podanej ścieżki i czyni podany katalog bieżącym dla dalszych operacji dyskowych.

Parent
Działa tak jak w AmigaDOS. Powoduje zmianę katalogu bieżącego na znajdujący się wyżej w hierarchii.

Dfree
Funkcja przyjmująca wartość równą ilości wolnych bajtów na dysku.

Print Dfree
Wyświetla na ekranie ilość wolnego miejsca na dysku.

Mkdir a$
a$ - Ścieżka dostępu i nazwa tworzonego katalogu.
Tworzy na dysku katalog o podanej nazwie. Np.:
Mkdir "df0:amiga"
Po wykonaniu tej instrukcji na dysku w stacji df0: utworzony zostanie katalog o nazwie "amiga".

Kill a$
a$ - Ścieżka dostępu i nazwa pliku do skasowania.
Kasuje z dysku plik o podanej nazwie.

Rename a$ To b$
a$ - Stara nazwa pliku.
b$ - Nowa nazwa pliku.
Zmiana nazwy podanego pliku.

Exist(a$)
a$ - Nazwa poszukiwanego pliku, katalogu, itp.
Funkcja ta przybiera wartość "-1", gdy podany plik lub katalog zostanie odnaleziony i "0" (zero), gdy dany plik nie istnieje. Instrukcja ta jest często używana do sprawdzania podłączenia dodatkowej stacji dysków.
Print Exist("Df1:")
Jeśli stacja jest podłączona otrzymamy wynik -1, a jeśli nie 0.

Fsel(a$,b$,c$,d$)
a$ - Ścieżka dostępu katalogu.
b$ - Tytuł pliku, który ma zostać wstępnie podświetlony.
c$ i d$ - Opcjonalne teksty informacyjne jakie mogą być wyświetlone w górnej części okna.
Jest to funkcja pozwalająca na wykorzystanie we własnym programie standardowego okna przeglądania dysku. Przyjmuje wartość pełnej ścieżki dostępu wybranego pliku. Jeśli żaden plik nie zostanie wybrany, zawiera ciąg pusty. Jej działanie dobrze jest zilustrować przykładem:

```
Rem Kasowanie plików
Do
F$=Fsel$("df0:"," ","Wybierz plik do skasowania")
If F$="" Then Edit
Kill F$
Loop
End
```

Programik ten wyświetla standardowy requester i kasuje z dysku wybrany plik. Radzę więc uważać przy jego używaniu, aby w "zapale twórczym" nie wykasować sobie potrzebnych danych.

Amos umożliwia także tworzenie własnych plików na dysku. Tym razem przedstawię zasady pracy z plikami sekwencyjnymi. Ogólnie można przyjąć, że przed rozpoczęciem operacji dyskowych należy plik "otworzyć", wykonać zapis lub odczyt danych i plik zamknąć. Przeznaczonych do tego jest kilka specjalnych komend.

Open Out kanał,a$
kanał - Numer kanału (od 1 do 10).
a$ - Tytuł pliku.
Instrukcja ta otwiera plik o podanej nazwie do zapisu. Jeśli plik taki już istnieje, zostaje skasowany i utworzony na nowo.
"Kanał" jest liczbą przeznaczoną do późniejszej identyfikacji pliku przy przesyłaniu danych.

Append kanał,a$
kanał - Numer kanału (od 1 do 10).
a$ - Tytuł pliku.
Działa podobnie jak Open Out, z tą różnicą, że gdy podany plik już istnieje, nie zostanie skasowany, a przesyłane dane zostaną dołączone na jego końcu.

Open In kanał,a$
kanał - Numer kanału (od 1 do 10).
a$ - Tytuł pliku.
Komenda ta otwiera plik do odczytu. Pozostałe parametry jak wyżej.

Close a
a - Numer kanału "na" którym jest otwarty plik.
Instrukcja ta zamyka obrabiany plik. Jest to konieczne, gdyż dane wysłane do pliku zostaną utrwalone dopiero po jego zamknięciu.
"a" to numer kanału jaki został podany przy otwieraniu pliku.

Print #kanał,a$,b$ ......
kanał - Numer podany przy otwieraniu pliku.
a$, b$ - Zmienne przeznaczone do umieszczenia w pliku.
Instrukcja ta działa podobnie jak Print, tylko, że nie wyświetla danych na ekranie, a wysyła je do pliku otwartego "na" podanym kanale.

Input #kanał,a$,b$ ......
kanał - Numer podany przy otwieraniu pliku.
a$, b$ - Zmienne, którym mają być przyporządkowane dane odczytane z pliku.
Komenda analogiczna do Input, ale zamiast z klawiatury odczytuje dane z pliku otwartego "na" podanym kanale. Wartości przypisywane kolejnym zmiennym muszą być oddzielone przecinkami i zakończone znakiem końca linii.

Input$(kanał,a)
a - Liczba odczytywanych znaków.
kanał - Kanał "na" którym otwarty jest plik.
Funkcja ta pozwala na odczytanie okeślonej liczby znaków z pliku.

Eof(kanał)
kanał - Kanał "na" którym otwarty jest plik.
Funkcja, która przyjmuje wartość "-1", gdy podczas odczytywania pliku osiągnięty zostanie jego koniec. Normalnie ma wartość 0.

Lof(kanał)
kanał - Kanał "na" którym otwarty jest plik.
Funkcja przyjmująca wartość równą długości pliku.

Pof(kanał)
kanał - Kanał "na" którym otwarty jest plik.
Funkcja ta pozwala na ustalenie miejsca odczytywania lub zapisywania pliku. Np.:
Pof(1)=10
Ustala pozycję na 10 znaków za początkiem pliku otwartego na kanale nr 1.

Teraz już czas na przykłady dotyczące odczytywania i zapisywania plików sekwencyjnych.

```
Open Out 1,"df0:test"
Print #1,"Amiga"
Close 1
```

Programik ten tworzy na dysku w stacji df0: plik o nazwie "test" i umieszcza w nim tekst "Amiga". Plik ten można odczytać dowolnym edytorem tekstu lub ponownie wykorzystać w programach odczytując go na przykład tak:

```
Open In 1,"df0:test"
Input #1,a$
Close 1
Print a$
```

Myślę, że w tej chwili jest już wszystko jasne i mogę przedstawić nieco większy program. Pokazuje on jeden ze sposobów tworzenia pliku konfiguracyjnego. Zgrywanie konfiguracji jest to często spotykane w programach użytkowych. Oto ten program:

```
Rem Tworzenie pliku konfiguracyjnego
Rem
Screen Open 0,640,256,16,Hires
Global A$,B$,C$,D$,E$
Paper 0
Clw
If Exist("test.config")=-1
Open In 1,"test.config"
Input #1,A$
Input #1,B$
Input #1,C$
Input #1,D$
Input #1,E$
Close 1
Else
A$="20"
B$="20"
C$="70"
D$="20"
E$="5"
End If
Do
A=Val(A$)
B=Val(B$)
C=Val(C$)
D=Val(D$)
E=Val(E$)
Wind Open 1,A,B,C,D,E
Paper 1
Clw
Print "To jest okno testowe."
Window 0
Locate 0,25
Centre "Czy chcesz wprowadzić nowe parametry okna ?"
Locate 0,27
Print "Lewy klawisz myszki  TAK"
Print "Prawy klawisz myszki  NIE"
Do
If Mouse Key=2 Then Edit
If Mouse Key=1 Then SET_CONFIG : Exit
Loop
Loop
End
Procedure SET_CONFIG
Window 1
Wind Close
Clw
Cdown
Centre "Podaj współrzędne graficzne lewego górnego rogu okna."
Locate 0,3
Input "Współrzędna X ";A$
Input "Współrzędna Y ";B$
Locate 0,8
Centre "Podaj szerokość i wysokość okna w liczbie wyświetlanych znaków."
Locate 0,11
Input "Szerokość okna ";C$
Input "Wysokość okna ";D$
Locate 0,16
Centre "Wybierz rodzaj ramki podając liczbę od 0 do 16 ('0' to okno bez ramki)."
Locate 0,19
Input "Numer ramki ";E$
Locate 0,25
Pen 3
Centre "Czy chcesz zapisać te dane na dysku?"
Pen 2
Locate 0,27
Print "Lewy klawisz myszki  TAK"
Print "Prawy klawisz myszki  NIE"
Do
If Mouse Key=1 Then Exit
If Mouse Key=2 Then Clw : Pop Proc
Loop
Open Out 1,"test.config"
Print #1,A$
Print #1,B$
Print #1,C$
Print #1,D$
Print #1,E$
Close 1
Clw
End Proc
```

Po jego wpisaniu i uruchomieniu na ekranie, pojawi się okno o wymiarach zdefiniowanych przeze mnie. Jeśli jednak zostaną podane nowe parametry i zapisze się je na dysku, przy każdym następnym uruchomieniu program przyjmie parametry podane przez użytkownika. Jest to możliwe dzięki plikowi "test.config" tworzonemu w bieżącym katalogu po zapisaniu "nastaw" na dysku. Przy uruchamianiu program poszukuje tego pliku na dysku i jeśli go znajdzie, wyświetla okno według parametrów w nim zawartych. Jeżeli go nie znajdzie, przyjmowane są "nastawy" zadeklarowane w programie. Analiza tego przykładu z pewnością pomoże wszystkim we wprowadzeniu podobnej funkcji w swoich programach. Jeśli ktoś ma ochotę może nawet użyć podanej tu procedury. Radzę jednak napisać własną, być może lepszą.

I to by było tym razem wszystko na temat współpracy programów amosa ze stacją dysków. Można jeszcze tworzyć pliki o bezpośrednim dostępie i zapisywać dane w poszczególnych rekordach, ale jest to zagadnienie dosyć obszerne i opiszę je przy okazji odcinka o tworzeniu baz danych, do którego lektury zapraszam w jednym z kolejnych numerów AMIGOWCA. W następnym odcinku naszego kursu trochę "pomuzykujemy", czyli wszystko o dźwięku w AMOSIE (efekty dźwiękowe, wykorzystanie gotowych modułów, itp.). ❑

AREXX cz. 2

Izabela Skibińska

W drugim odcinku naszego cyklu o Arexxie będziemy kontynuować zapoznawanie się ze składnią Arexxa, wgłębimy się w sens paru ważnych i mocnych rozkazów tego języka, a także zajmiemy się rexxsupport.library - jedyną biblioteką dodatkową dostarczaną wraz z całym środowiskiem.

Po lekturze tego artykułu możemy uznać, że elementy Arexxa związane ze składnią mamy już opanowane. Pozostaje już tylko zapoznanie się z rozkazami nieskładniowymi (nie sterującymi przebiegiem programu) oraz z funkcjami "wbudowanymi".

Instrukcja "Parse"

Na początek prościutki programik ilustrujący działanie bardzo ważnego rozkazu PARSE, którego suchy opis w podręczniku może nie dać od razu pełnego zrozumienia.

Listing 1

```
/* Ilustracja działania rozkazu PARSE */
/* Dodawanie dwóch liczb */

    options prompt  "Dodaj dwie liczby>"
    do forever
    parse upper pull answer
            IF answer = "Q" | answer = '' then leave
            parse var answer  liczba1'+'liczba2
            parse numeric dig f w
            if dig=9 then numeric digits 6
            say liczba1
            say liczba2
            say liczba1+liczba2
     end

parse source function komfun  nazwa path ext host
          IF komfun = 1  then
                   SAY "Program " nazwa"."ext "pracowal jako funkcja"
                            else
                   SAY  "Program "nazwa"."ext "pracowal jako rozkaz"

parse version arexx wersja cpu fpu video freq

parse value "Informacje o środowisku pracy programu "nazwa".
"ext with inf
say inf
say "Host address"  host
say "Wersja Arexxa" strip(wersja,'L','V')
say "Procesor" cpu
say "Koprocesor "mpu
say "Standard video "video
```

UWAGA! Tekst wytłuszczony należy wpisywać w jednej linii.

Definicja rozkazu PARSE wygląda następująco:

PARSE[UPPER] string string1 string2 ......

Ogólnie rzecz ujmując rozkaz PARSE służy do manipulacji ciągiem znaków (string), który może pochodzić z wejścia standardowego (błędów też) lub być zawarty w zmiennej. Słowa kluczowe określają nam sposób manipulacji stringiem. Wyniki analizy umieszczane są w ciągach string1, string2... Zależy na ile części i jak rozbijamy analizowany ciąg znaków.

Oto ich pełny zestaw:
UPPER - string zostanie zamieniony na duże litery. Musi być pierwszym słowem kluczowym występującym po rozkazie.
ARG - ciąg znaków bądź ciągi znaków zostaną pobrane z argumentu, z którym wywołany był program.
EXTERNAL- ciąg znaków będzie pobrany z STDERR (standardowego wejścia błędów).
NUMERIC - Zostanie dokonana analiza opcji numerycznych. Aby dostać się do tych wartości konieczne jest rozbicie na trzy zmienne. (W przykładzie: dig f w). W kolejności oznaczają one: liczbę cyfr braną pod uwagę w obliczeniach, liczbę cyfr ignorowaną przy porównaniach, format wyprowadzanej liczby. NUMERIC jest także rozkazem, którym możemy ustawić te wartości. Kolejne słowa kluczowe potrzebne do ustawienia parametrów opisanych wyżej to DIGITS, FUZZ i FORM (Patrz przykład).
PULL - ciąg znaków zostanie pobrany z konsoli.
SOURCE - mamy możliwość uzyskania informacji o naszym programie. Czyli jak został wywołany (jako funkcja czy rozkaz), jak się nazywa, jego pełną ścieżkę dostępu, rozszerzenie oraz nazwę portu (host-adress), gdzie wysyłane są rozkazy. (Patrz przykład).
VALUE wyr WITH str - wyrażenie wyr będące dowolną konkatenacją zmiennych i stringów zostanie zapamiętana pod zmienną str. (Patrz przykład).
VAR - ciąg znaków zostanie pobrany ze zmiennej poprzedzonej tym słowem kluczowym. Ciąg może być dzielony w dowolny sposób. Można uznać, że jest to możliwość odwrotna w stosunku do poprzedniej. (Patrz przykład). Znając położenie interesujących nas znaków możemy rozbijać ciąg podając pozycje znaku albo przesunięcie (z "+" lub "-") w stosunku do poprzedniego położenia. Np:

```
zm="123456789"
parse var zm 2 a +2 b -3 c
say a
say b
say c
spowoduje wyprowadzenie napisu:
23
456789
123456789
```

VERSION - analogicznie jak przy opcjach NUMERIC i SOURCE. Tym razem otrzymujemy informację o cpu, fpu, wersji Arexxa, standardzie video, itp.

Instrukcje sterujące

W Listingu 1 występuje konstrukcja typu "IF wyr THEN I1", gdzie "wyr" jest wyrażeniem przyjmującym wartość logiczną. Po THEN musi występować instrukcja lub blok instrukcji (blok instrukcji zawieramy między słowami kluczowymi DO i END, o czym była mowa w części pierwszej kursu), które są wykonywane, gdy wartość logiczna wyrażenia jest prawdziwa (przyjmuje wartość 1). Gdy chcemy podjąć jakieś akcje w przypadku przyjęcia przez wyrażenia logiczne wartości 0 (fałsz), używamy

słowa kluczowego ELSE. Pełna postać instrukcji wygląda więc następująco:

IF wyr THEN I1 ELSE I2
gdzie:
I1,I2 to instrukcje lub bloki instrukcji.

Przy potrzebie rozpatrzenia większej ilości warunków, zamiast gubić się w gąszczach IF THEN i ELSE, wygodniej jest posłużyć się instrukcją SELECT (Listing 4) Między rozkazami SELECT i END instrukcjami "WHEN wyrn THEN In" n=1,2,... możemy rozpatrywać dowolną liczbę warunków i podejmować odpowiednie akcje. Słowo kluczowe OTHERWISE jest opcjonalne. Gdy żaden z warunków nie zostanie spełniony zostaną wykonane instrukcje po OTHERWISE. Przy rozpatrywaniu wielu warunków, czasami wygodne jest korzystanie z rozkazu NOP, czyli nie rób nic. Przykład pełnej wersji instrukcji SELECT znajduje się w listingu 4, gdy analizowana jest odpowiedź użytkownika. Przy tworzeniu wyrażeń logicznych określających warunki bądź też wyrażeń arytmetycznych, używane są różnego rodzaju operatory. Pełne ich zestawienie wraz z priorytetami znajduje się w tabelkach.

W Listingu 1 występuje rozkaz LEAVE. Jak można się domyślać powoduje on opuszczenie pętli. Istnieje także rozkaz ITERATE, dzięki któremu możemy przejść do następnej iteracji. Zarówno po LEAVE jak i po ITERATE, możemy umieścić zmienną sterującą pętli. Przydatne jest to wtedy, gdy chodzi nam o pętle zewnętrzne w stosunku do tej, w której jesteśmy.

Przykład:

```
do i=1 to 2
        do j=1 to 2
         if i=j then iterate i
         else
         say "i="i" j="j
        end
       end
```

Wyprowadzony zostanie napis "i=2 j=1".

Natomiast, gdy chcemy wyjść z bloku instrukcji używamy rozkazu BREAK.

Przykład:

```
do
say "Hello1"
break
say "Hello2"
end
```

Napis "Hello2" nie zostanie wyprowadzony.

Za kolejną instrukcję sterującą możemy uznać EXIT [n]. Powoduje ona natychmiastowe wyjście z programu z kodem określonym przez n.

Interakcja z AmigaDOS-em

Ostatni trochę dłuższy program Arexxowy (Listing 4) ilustruje między innymi możliwość stosowania tego języka jako środka służącego do tworzenia prostych programów narzędziowych oraz jako rozszerzenie AmigaDOS-u. Z poziomu Arexxa mamy bowiem dostęp, poprzez komendę ADDRESS COMMAND, do wszystkich rozkazów AmigaDOS-u. (Do komendy ADDRESS jeszcze wrócimy). Ta interakcja z AmigaDos-em daje bardzo duże możliwości skryptowe. Wyobraźmy sobie, że mamy dwa bardzo długie katalogi. Niektóre pliki w katalogu pierwszym są zdublowane w drugim, a my chcemy, aby dwa katalogi nie miały jednakowych plików. Automatyzacja tej operacji z poziomu AmigaDos-u jest raczej niemożliwa. W Arexxie wystarczy prosty skrypt:

Listing 2

```
  /* Kasowanie z kat2 powtarzających się plików z kat1 */

address command "list ram:kat1 > T:temp1 NOHEAD NODATES"
address command "list ram:kat2 > T:temp2 NOHEAD NODATES"
   i=open('handle1','T:temp1','R');j=open('handle2','T:temp2','R')
    if (i=0 | j=0) then exit 10
do until (eof('handle1'))
   string1 = readln('handle1')
   string1 = subword(string1,1,1)
      do until (eof('handle2'))
        string2 = readln('handle2')
        string2 = subword(string2,1,1)
       if (compare(string1,string2)=0) then
            do
        if string1~=""  then
         address command "delete ram:kat2/"string1
             leave
        end
  end
end

 close('handle1')
 close('handle2')
```

Procedury i funkcje

Zanim przejdziemy do analizy programu z listingu 4 parę słów o tworzeniu i własnościach procedur. Zdefiniowanie podprogramu w Arexxie polega na podaniu jego nazwy, po nazwie umieszczamy dwukropek, a następnie opcjonalnie słowo kluczowe PROCEDURE. Każdą procedurę kończymy słowem kluczowym RETURN. Procedurę wywołujemy podając jej nazwę. Możemy ją poprzedzić słowem kluczowym CALL. Po nazwie, w nawiasach okrągłych, możemy przesłać listę argumentów oddzielonych przecinkami, które w "ciele" funkcji przechwytujemy funkcją arg(n) (n-numer argumentu). Argumentami mogą być wyrażenia bądź funkcje zwracające wartość.

Procedurom w momencie wywołania zostaje przydzielony obszar na własne zmienne lokalne (a więc możliwość stosowania rekurencji), gdzie znajdują się między innymi kopie zmiennych środowiskowych z zainicjalizowanymi wartościami obowiązującymi przed wywołaniem funkcji (np. NUMERIC DIGITS, czy też wzorzec promptu ustawiany rozkazem OPTIONS PROMPT). Nie podając słowa kluczowego PROCEDURE mamy możliwość odwołania się do zmiennych z programu głównego. Natomiast podając ten rozkaz, możemy opcją EXPOSE wyselekcjonować zmienne z programu głównego, które chcemy "przykryć". Arexx nie umożliwia definiowania procedur wewnątrz procedur, jak to jest w przypadku Pascala. Zwróćcie uwagę na słowo kluczowe EXIT stojące przed definicjami procedur. Bez niego interpreter wyrzuciłby błąd. Wszystko wyjaśnić powinien listing 3.

Listing 3

```
/* Procedury */

  options prompt ">>>"
  pull i ; pull j

  say 'i = ' i 'na poziomie głównym '
  say 'j = ' j 'na poziomie głównym'

  call Proc1
  say 'i = 'i 'niezmienione  po wyjściu z Proc1'
  say 'j = 'j 'zmienione  po wyjściu z Proc1 '
  call Proc2
  say 'i = 'i 'zmienione  po wyjściu z Proc2'
  say 'j = 'j 'zmienione  po wyjsciu z Proc2 '
  exit

   Proc1:procedure expose j
   options prompt ">"
    pull i: pull j:
       say 'i=' i  'w procedurze Proc1'
       say 'j= ' j 'w procedurze Proc1'
    return


   Proc2:
   options prompt ">>"
   pull i;pull j:
      say 'i=' i  'w procedurze Proc1'
      say 'j= ' j 'w procedurze Proc1'
   return
```

W Listingu 4 zastosowano rekurencyjną funkcję viewdir() do przeglądania katologu dyskietki. Funkcja zapisuje w pliku tymczasowym sformatowane, w zależności od zagłębienia rekurencji, drzewo katalogu dyskietki.

Arexx posiada dosyć duży wybór funkcji wbudowanych. Zawsze lepiej, gdy tylko jest to możliwe, wykorzystywać gotowe funkcje, przynajmniej dlatego, że są one już zinterpretowane. Z grubsza możemy dokonać pewej systematyzacji tych wbudowanych podprogramów.

1. Funkcje do obsługi plików: (Open(), Close(), Eof(), Seek(), Exists(), Writeln(), Readln(), Readch(), Writech()).
2. Funkcje do operacji na bitach: (Bitchg(), Bitclr(), Bitcomp(), Bitset(), Bittst(), Bitxor()).
3. Funkcje operujące na pamięci: (Storage(), Getspace(), Freespace(), Import(), Export()).
4. Funkcje operujące na stringach i ciągach stringów oddzielonych spacjami. Można podzielić je na 3 podgrupy:

a. modyfikujące częściowo dany string: (Insert(), Compress(), Center(), Copies(), Delstr(), Insert(), Left(), Overlay(), Reverse(), Right(), Space(), Strip(), Substr(), SubWord(), Translate(), Trim(), Upper(), Delword(), Word()),
b. modyfikujące cały string - różnego rodzaju konwersje: (X2C(), X2D(), D2C(), D2X(), C2D(), C2X(), Xrange()),
c. zwracające wartości liczbowe dotyczące operacji na stringach: (Abbrev(), Compare(), Index(), Lastpos(), Length(), Lines(), Pos(), Words(), Wordlength(), Wordindex()).

5. Funkcje dotyczące systemu i środowiska Arexxa: (Time(), Show(), Sourceline(), SetClip(), Remlib(), Pragma(), GetClip(), Addlib(), Address(), Trace()).
6. Pozostałe (już bardzo niewielka ilość).

We wszystkich naszych przykładach korzystaliśmy z różnych funkcji wbudowanych. Nie będziemy jednak ich opisywać, gdyż zajęłoby nam to zbyt dużo miejsca, a ponadto opis taki jest ogólnie dostępny. Przyjrzymy się natomiast funkcjom z rexxsupport.library.

Rexxsupport.library

Jak już wspominałam w poprzednim odcinku, Arexx jest systemem otwartym i w związku z tym nieustannie powstają dla niego biblioteki, zwiększające jego możliwości w zakresie interakcji z całym systemem operacyjnym Amigi. Powstały już biblioteki umożliwiające mu korzystanie z Intuition, graphics.library, a ostatnio nawet z bardzo popularnej reqtools. library. Gdybyśmy zaopatrzyli program z Listingu 4 w cały interface Intuition oraz wykorzystali requestery z rexxreqtools.library, wyglądałby tak jak "normalny" program użytkowy.

W Listingu 4 wykorzystałam natomiast rexxsupport.library, a dokładniej jej funkcje showdir(), do stworzenia listy plików i katalogów. Nie trzeba więc tworzyć tymczasowego pliku jak w Listingu nr 2. Aby skorzystać z biblioteki Arexxowej należy najpierw umieścić ją na liście bibliotek systemowych. Służy do tego funkcja Arexxowa AddLib. Jej argumentami są w kolejności: nazwa biblioteki, priorytet, offset i minimalna wersja. Priorytet i minimalna wersja jaka nas interesuje mogą być ustawione na zero. Offset jest wartością charakterystyczną dla danej biblioteki i nie ma znaczenia dla użytkownika. W przypadku rexxsupport.library wynosi on -30. Istnieje także możliwość dołączenia biblioteki do środowiska Arexxa spod Shella poprzez wywołanie programu rxlib z argumentami identycznymi jak przy funkcji AddLib.

Biblioteka Rexxsupport pozwala na wykorzystanie takich mechanizmów systemu jak alokacja pamięci, tworzenie message portów, odbieranie message'ów. Aby zarezerwować pamięć wywołujemy funkcję AllocMem:
base = AllocMem(N,[A])
Argumenty:
N - wielkość pamięci w bajtach.
A - rodzaj pamięci (podajemy standardowe wartości rozumiane przez EXEC).
Wynik: base-ciąg określający początek zaalokowanej pamięci.
Zwalniamy pamięć funkcją FreeMem.
ret = FreeMem(base,N)
Wynik: 1- w przypadku powodzenia, 0 w przypadku przeciwnym.
Funkcją Delay wstrzymujemy działanie programu na określony czas.
Delay(ticks)
ticks - czas w 1/50 sekundy.

Przykład

```
public     = '00 00 00 01'x
chip = '00 00 00 02 'x
fast = '00 00 00 04'x
clear = '00 01 00 00 'x

membase = AllocMem(10000,bitor(chip,clear))
say "Zarezerwowano 10KB od adresu "c2x(base) " w CHIP
RAMie"
say "Czekamy 4 sekundy "
Delay(200)
FreeMem(membase,10000)
say "Pamięć znów wolna."
```

Ważną cechą rexxsupport.library jest zdolność tworzenia message portów, którym może zarządzać program. Istnieje więc możliwość nie tylko wysyłania message'ów do innych programów, ale też ich przyjmowania.
portbase=OpenPort(name)
Wynik: portbase - adres portu albo 0 w przypadku niepowodzenia
Funkcją getpkt() możemy pobrać message z portu.
mess = getpkt(name)
name - nazwa portu
mess - pobrana message
Zgodnie z tym co pisałam w poprzednim odcinku o komunikacji między procesami, gdy pobierzemy message z portu, należy na nią jak najszybciej "odpowiedzieć" i uwolnić system od czekania. Do tego celu używamy funkcji Reply().
Reply(mess,rc)
mess - "odsyłana" message
rc - kod powrotu, ustawiany na 0, gdy wszystko jest w porządku
Funkcją ShowList możemy przejrzeć listy systemowe.
ret=ShowList(typ,[name],[znak])
typ - mogą to być:
PORTS (porty),
LIBRARIES (biblioteki),
DEVICES (urządzenia), itp
name - gdy na liście jest "name" zwracana jest jako 1 lub 0 w przeciwnym przypadku znak - elementy listy odzielone są tym znakiem, a nie spacją
ret - lista lub odpowiednia wartość liczbowa, gdy podano parametr name

Przykład:

```
portbase = OpenPort('MYPORT')
if portbase=0 then
  SAY "Nie można stworzyć portu."
 else
       do
        say "Port MYPORT jest."
        call ShowPorts
        ClosePort('MYPORT')
        say "                    "
        say "Portu MYPORT nie ma."
        call ShowPorts
        end
  exit
ShowPorts: procedure
porty = ShowList(PORTS,,';')
do until porty=""
   parse var porty port';'porty
   say port
 end
return
```

W listingu 4 zastosowałam funkcję ShowDir. Ma ona argumenty analogiczne do poprzedniej funkcji ShowList z tym, że chodzi tu oczywiście o pliki i katalogi. Jako seperatora celowo użyłam znaku zastrzeżonego w nazwach plików i katalogów - "/".

Listing 4

```
/* Program do tworzenia katalogu zawartości dyskietek */

message_col1 = '9b'x||'1;32;41m'
```

```
message_col2 = '9b'x||'1;33;40m'
reset = '9b'x||'0m'

 if ~show(Libraries,"rexxsupport.library ")
              then
              addlib("rexxsupport.library",0,-30,0)

drive = 'df0:'

say  message_col1 "Program do katalogowania dyskietek"
say "                                                                    "
say  message_col2 "Numer stacji 0-3 ?"reset

 do until (answer=0 | answer=1 | answer=2 | answer=3 | answer='' |
 answer='q' )
              pull answer
              select
               when answer=1 then drive="df1:"
               when answer=2 then drive="df2:"
               when answer=3 then drive="df3:"
               when answer=0 | answer = '' then nop
               when upper(answer)='Q' then
                              do
                                 say "Bye"
                                 exit
                              end
               otherwise say "Error"
               end
              end
 if ~Exists("ram:spis")  then
 address command  'makedir ram:Spis'

 address command  'assign SPIS: ram:spis'
 prevname = ''
 name = ''
 do forever
   do  while (name = prevname)
     say message_col2"Insert new disc and press any key when ready"reset
       pull answ
       address command "Info > T:tem1" drive
       call open('handle','T:tem1','R')
       do for 4
         line = readln('handle')
       end
       call close('handle')
       if words(line) = 9 then
         name = word(line,9)   /* Read Only */
        else
         name = word(line,8)   /* Read/Write */
   end
   address command 'delete T:tem1 quiet'
   prevname = name

  open('kat',"T:kat",'W')
  call viewdir(drive,0)
  call close('kat')
  address command 'copy T:kat  Spis:'name
  say message_col1 "Another Disk ? Y/N"reset
  pull answ
  if answ = 'N' then leave
end
address command 'assign Spis: remove'
say reset
exit

viewdir:    procedure
              parse arg dir,level
              pliki = showdir(dir,'F','/')
              do forever
                 parse var pliki  nast'/'pliki
                 if nast = '' then leave
               do for level
                  nast=insert('--',nast)
               end
               writel('kat',nast)
        end

      katalogi = showdir(dir,'D','/')
        do forever
          parse var katalogi pop'/'katalogi
          if  pop == '' then leave
          pop= insert(pop,'/')
          pop = dir||pop
           writeln('kat',pop)
          call viewdir(pop,level+1)
        end
  return
```

UWAGA!: Tekst wytłuszczony należy wpisywać w jednej linii.

Tabelka 1:
Operatory arytmetyczne.

Symbol	Priorytet	
+	8	unarny plus
-	8	unarny minus
**	7	potęgowanie
*	6	mnożenie
/	6	dzielenie
%	6	dzielenie całkowite
//	6	reszta z dzielenia całkowitego
+	5	dodawanie
-	5	odejmowanie

Tabelka 2:
Operatory logiczne.

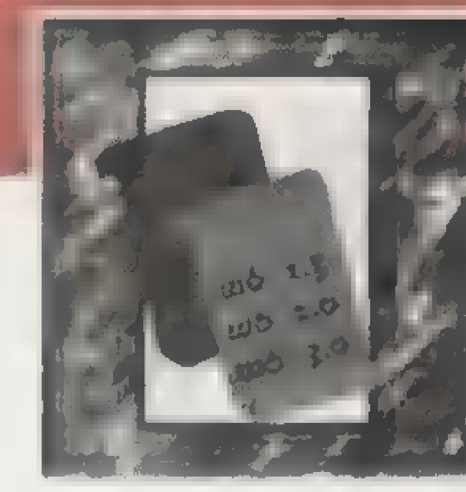
Symbol	Priorytet	Operacja
~	8	NOT
&	2	AND
|	1	OR
^ ub &&	1	XOR

Tabelka 3:
Operatory porównania.

Symbol	Priorytet	Operacja
==	3	identyczność (ciągi znakowe)
~==	3	nieidentyczność (ciągi znakowe)
=	3	równość
~=	3	nierówność
>	3	większe
>= lub ~<	3	większe lub równe
<	3	mniejsze
+	5	dodawanie
<= lub ~>	3	mniejsze lub równe

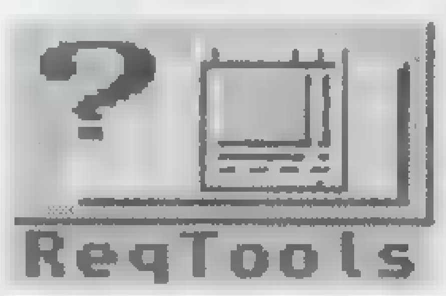
Tabelka4:
Operatory łączenia.

Symbol	Operacja
||	połączenie ciągów znakowych bez spacji między nimi
(spacja)	połączenie ciągów znakowych ze spacją między nimi



Praktyczne narzędzia cz.2

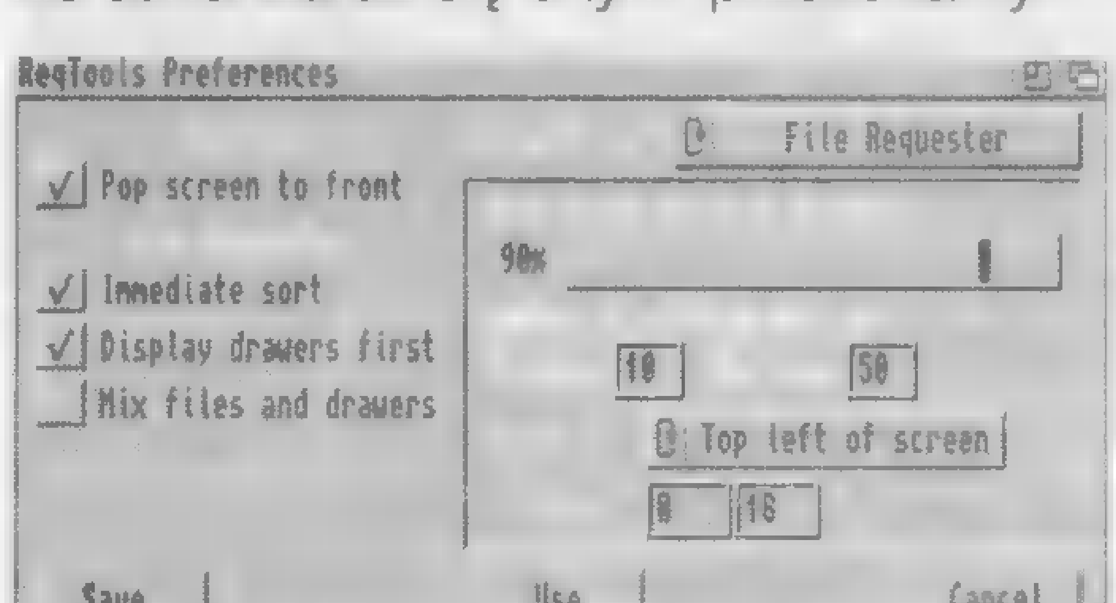
W poprzednim numerze Amigowca były zamieszczone opisy wybranych programów narzędziowych. Dzisiaj kolej na trzech następnych bohaterów: Req Toolsa, Tools Daemona i MKS_Lensa.



Pierwszy z nich - "Req Tools" związany jest z najbardziej popularną niesystemową biblioteką - "req.library". Starzy Amigowcy pamiętają zapewne, że towarzyszy ona Amidze prawie od samego początku jej istnienia. Żeby zrozumieć dlaczego odniosła tak duży sukces, należy sięgnąć do historii.

Pierwsze systemy, które powstały dla Amigi miały stosunkowo ubogie możliwości. Przede wszystkim brakowało im pewnego standardu okien dialogowych i gadżetów tak, jak to miało miejsce np. w Macintoshu. W ten sposób powstały idealne warunki dla rozwoju nowej biblioteki "req.library", która w znacznym stopniu wypełniała braki. Dzięki swojej przemyślanej strukturze zaczęła przyciągać więcej programistów, którzy coraz częściej zaczęli wykorzystywać jej zalety w swoich programach (np. Cygnus Edytor). Wreszcie wyłonił się pierwszy standard. W momencie gdy powstał system 2.0 oferujący nowe możliwości, nad "req.library" pojawiły się czarne chmury. Ustanowiony przez firmę Commodore jednolity standard zupełnie odbiegał od zasad przyjętych w "req.library", a poza tym był od niej o wiele lepszy. Spowodowało to, że programiści piszący pod systemem 2.0 przestali korzystać z "req.library". Wszystko byłoby dobrze, gdyby nie to, że bogata oferta starego oprogramowania nie pasowała do nowego środowiska. Dlatego też postanowiono zmodyfikować starą poczciwą "req.library" tak, aby była zachowana zgodność programowa, a zarazem szata graficzna pasowała do nowego standardu. W 1991 roku po jawiła się zmodyfikowana wersja tej bi blioteki. Dzięki niej wiele programów zyskało nowe oblicze i przeżyło swoją drugą młodość.

Nowa "req.library" może być uaktywniona na dwa sposoby. Pierwszy jest taki sam jak dla pozostałych bibliotek, czyli z "req.library" korzystają tylko te programy, które zostały na nią napisane. Drugi sposób jest pewnego rodzaju nowością. Polega on na tym, że po uruchomieniu specjalnego programu "RT-Path", wektory bibliotek systemowych "asl.library", "arp.library" i "intuition.library" (odpowiedzialnych za okna dialogowe), zostają zmienione w taki sposób, żeby wskazywały na "req.library". Nie trudno się domyślić, że program odwołujący się np. do "asl.library", w rzeczywistości będzie korzystać z "req.library" (chociaż wcale o tym nie będzie wiedział). Nasuwa się pytanie, czy ma to jakiś sens. Zaraz się o tym przekonamy.

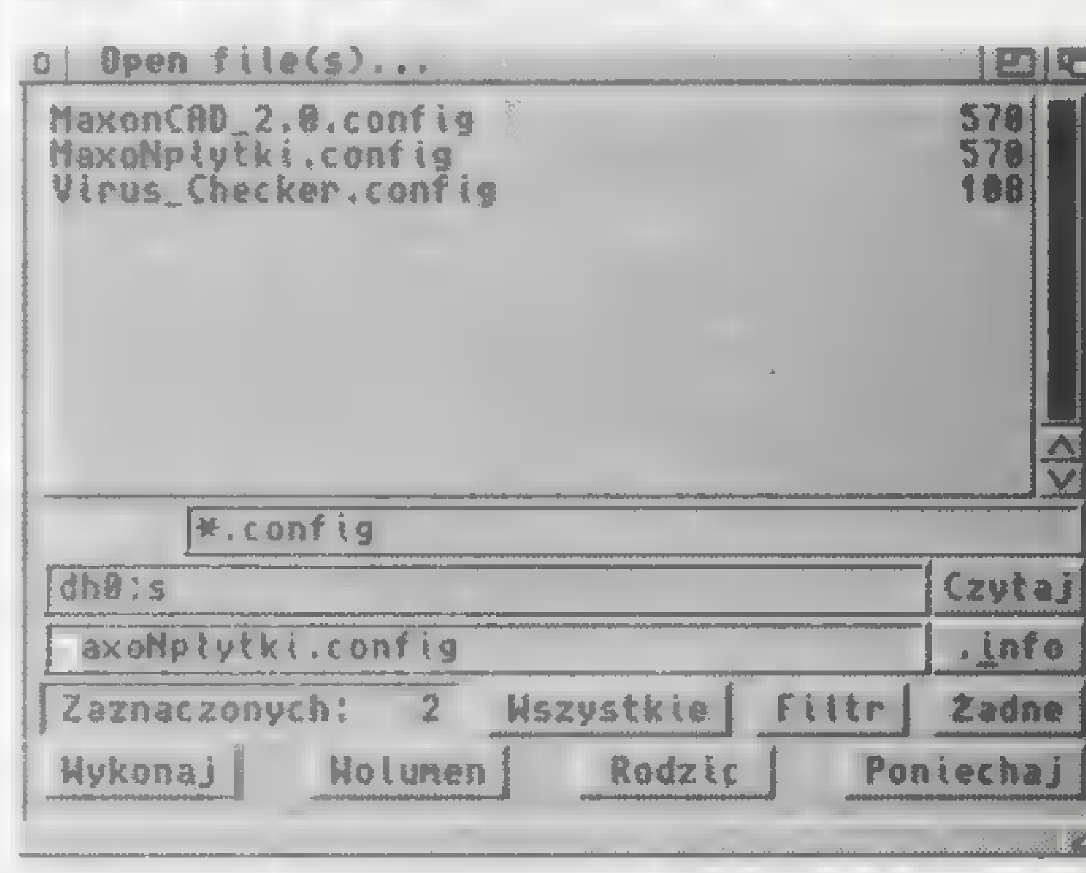


Rysunek 1.

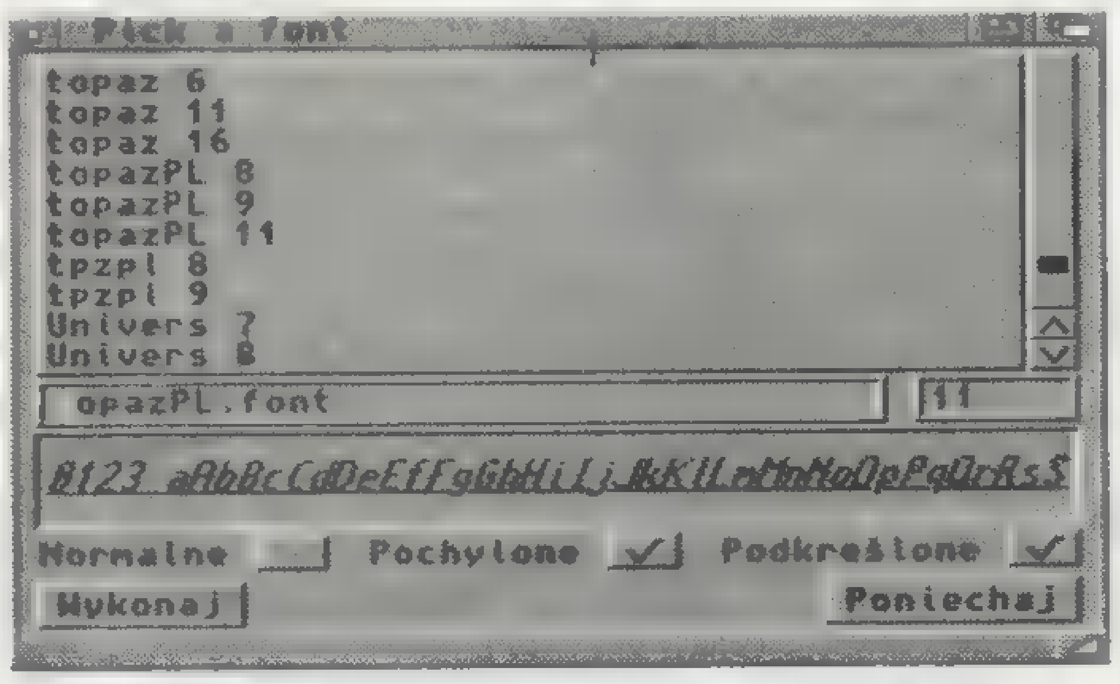
Biblioteka "req.library" jest obsługiwana przez specjalny program "Req-Tools" umieszczony wraz z innymi preferencjami w szufladzie "Prefs" (Rysunek 1). Przy jego pomocy można skonfigurować okna dialogowe według własnych upodobań. Możliwe jest przy tym określenie np. wymiarów okien z komunikatami w procentach w stosunku do całego ekranu, albo wyznaczenia miejsca, w którym będą się one pojawiać. Bardzo ciekawy efekt można uzyskać, gdy jako miejsce ukazywania wybierze się kursor myszki. Nie trzeba wtedy "latać" po całym ekranie, żeby kliknąć na odpowiedni gadżet. Poza tym interesująca jest różnica pomiędzy opcjami np. "Top left of screen", a "Top left on window". Pierwsza powoduje, że okna dialogowe będą ukazywały się w lewym górnym rogu ekranu, a druga, że w lewym górnym rogu okna. Jest to szczególnie przydatne przy pracy z wysokimi rozdzielczościami. Kolejnym parametrem, który można ustawiać, jest sposób sortowania: najpierw katalogi, a potem pliki, najpierw pliki, a potem katalogi, albo wszystko pomieszane razem wg alfabetu.

Ostatnia wersja biblioteki "req.library" ma zdefiniowane następujące rodzaje gotowych okien dialogowych:

1) do wskazywania plików, katalogów lub wolumenów,

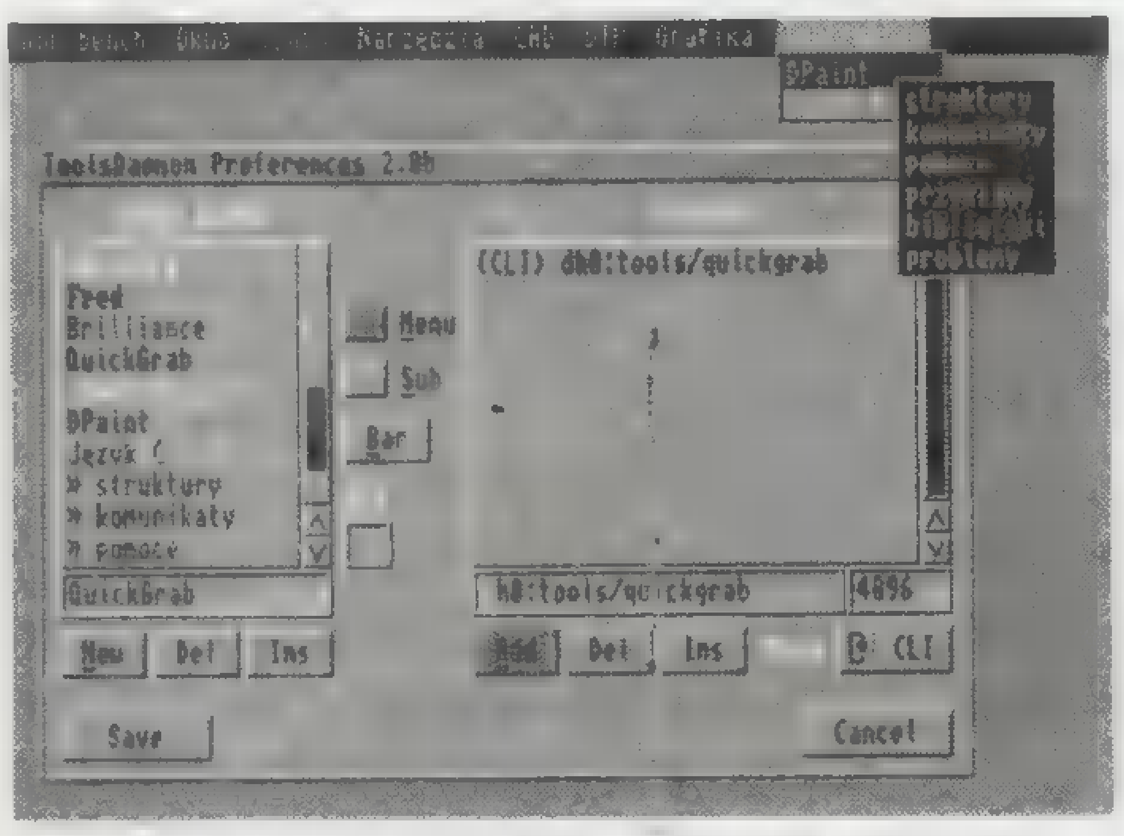


2) do wybierania fontów,



3) do definiowania palety kolorów,

4) do określania trybu graficznego ekranu,



5) do wczytywania z klawiatury tekstów i liczb,

6) do wyświetlania wszelkiego rodzaju komunikatów.

Oprócz tego każdy rodzaj posiada wiele swoich odmian - czyli w rzeczywistości istnieje kilkadziesiąt różnych okien dialogowych (i nie tylko dialogowych). Wraz z programem dostarczana jest obszerna instrukcja, w której zainteresowani czytelnicy mogą znaleźć wszystkie niezbędne informacje dotyczące tego tematu łącznie z zasadami programowania.

Jak wspomniałem wcześniej, nieodłączną częścią "req.library" jest program "RTPath", który służy do zmiany wektorów bibliotek systemowych. Patcher musi być uruchamiany z odpowiednimi paramertami. W zależności od tego, które biblioteki mają zostać "podmienione", należy wpisać:

ASL lub NOALS,

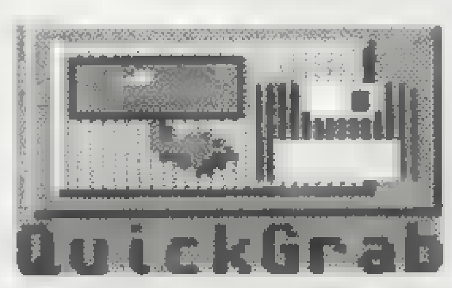
ARP lub NOARP,

INTUITION lub NOINTUITION.

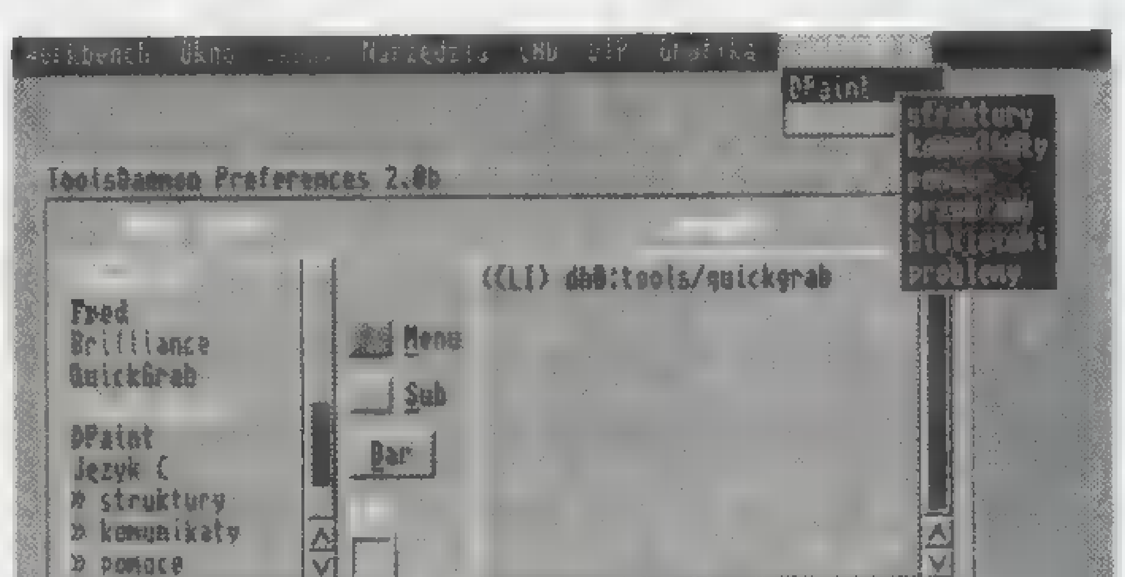
Poza tym można spowodować, że podczas uruchamiania nie będą ukazywały się żadne komunikaty. Jest to szczególnie pożądane, gdy program wgrywa się automatycznie (np. z szuflady WBStartup).

Podsumowując można śmiało stwierdzić, że dzięki nowej "req.library" wiele starych programów może przeżyć swą drugą młodość (niektórzy powiedzą, że jest w tym trochę przesady). Poza tym dzięki wprowadzeniu możliwości ustawiania preferencji dla okien dialogowych, program doskonale uzupełnia system. Instalacja na dysku systemowym jest prosta (dołączony jest program instalacyjny), a całość nie zajmuje dużo miejsca.

Kolejne narzędzie na które chciałbym zwrócić uwagę to:



Służy ono do definiowania menu użytkownika dostępnego z poziomu Workbencha (Rysunek 2). Podobny program - "ToolManager" - jednak o znacznie większych możliwościach, był już opisywany w zeszłym numerze Amigowca. Pomimo że obu programom przyświeca ten sam cel, trudno je ze sobą porównywać. Podstawowa różnica między nimi tkwi w sposobie realizacji zadania. "ToolManager" nastawiony jest przede wszystkim na jak największą ilość możliwych sposobów uruchamiania programów (zwykłe ikony, ikony dotykowe, menu...), co czyni go trochę zagmatwanym. Natomiast ToolDaemon wyspecjalizowany jest tylko i wyłącznie w menu, ale za to oferuje bardziej wyrafinowane możliwości jego rozbudowy.

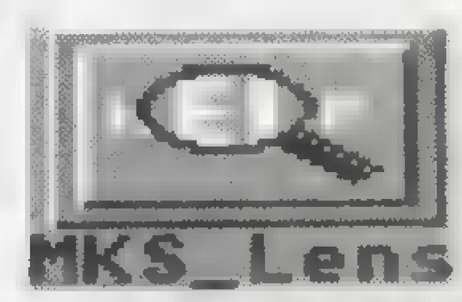


Rysunek 2.

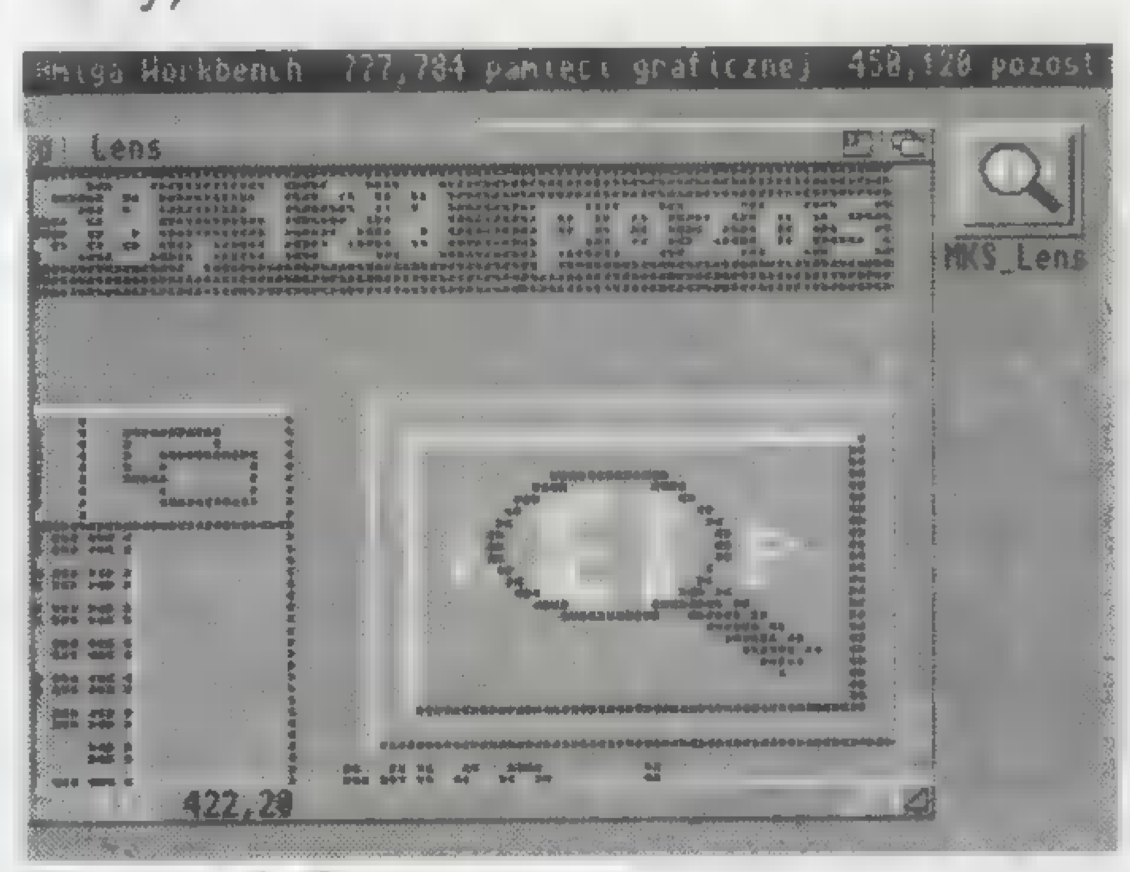
Jego zaletą jest to, że umożliwia tworzenie budowy hierarchicznej menu. Przykład takiego rozwiązania przedstawiony jest na Rysunku 2. Dzięki temu możemy umieścić o wiele więcej pozycji w pojedynczym menu nic nie tracąc na jego przejrzystości. Poza tym pomiędzy kolejnymi opcjami w menu można umieszczać specjalne wężyki (Bar), które mogą np. oddzielać grupy opcji uszeregowanych tematycznie. Pod każdą opcję w menu można wpisać dowolną ilość uruchamianych komend, z zaznaczeniem sposobu zasymulowania ich uruchomienia (z Workbencha albo z CLI).

Projektowanie menu w programie ToolDaemon jest bardzo łatwe i przyjemne. Każdą pozycję można dowolnie przesuwać w górę i w dół. Jest to niewątpliwą zaletą tego programu. Znacznie gorzej wygląda wpisywanie komend, które mają być uruchamiane. Brak możliwości wybierania plików z okna dialogowego znacznie utrudnia wykonanie zadania. Wszystkie teksty trzeba wpisywać z klawiatury bez możliwości sprawdzenia ich poprawności. Pomimo tej dość znacznej wady uważam, że ToolDaemon doskonale spełnia swoje zadanie i zasługuje na dobrą ocenę.

Na koniec chciałbym zaprezentować jeszcze jedno małe narzędzie (może bardziej zabawkę).



Jest to lupa, która w specjalnym oknie powiększa wszystko to, co znajduje się pod kursorem myszki (Rysunek 3). Może być zastosowana wszędzie tam, gdzie trzeba dokładnie ustawiać jakieś obiekty (np. ikony).



Rysunek 3.

Jak na lupę komputerową przystało posiada kilka stopni powiększania. Może być uruchamiana na dowolnym ekranie i swobodnie "przeskakiwać" pomiędzy nimi. Działa bardzo szybko i wbrew pozorom prawie wcale nie zwalnia komputera.

Kolejna porcja nowych narzędzi już wkrótce. ❑

REPRO STUDIO UNIVERSAL

Krzysztof Wirszyłło

Na Amigę zawsze istniały dobre programy graficzne, choćby takie jak Deluxe Paint czy TV Paint. Były to jednak pakiety czysto malarskie, bez ukierunkowania na poligrafię, czyli na obróbkę zdjęć. Z czasem zaczęły powstawać odpowiednie narzędzia. Istnieją programy Image Master czy Image FX, lecz nadal brakuje programu na miarę Adobe Photoshop.

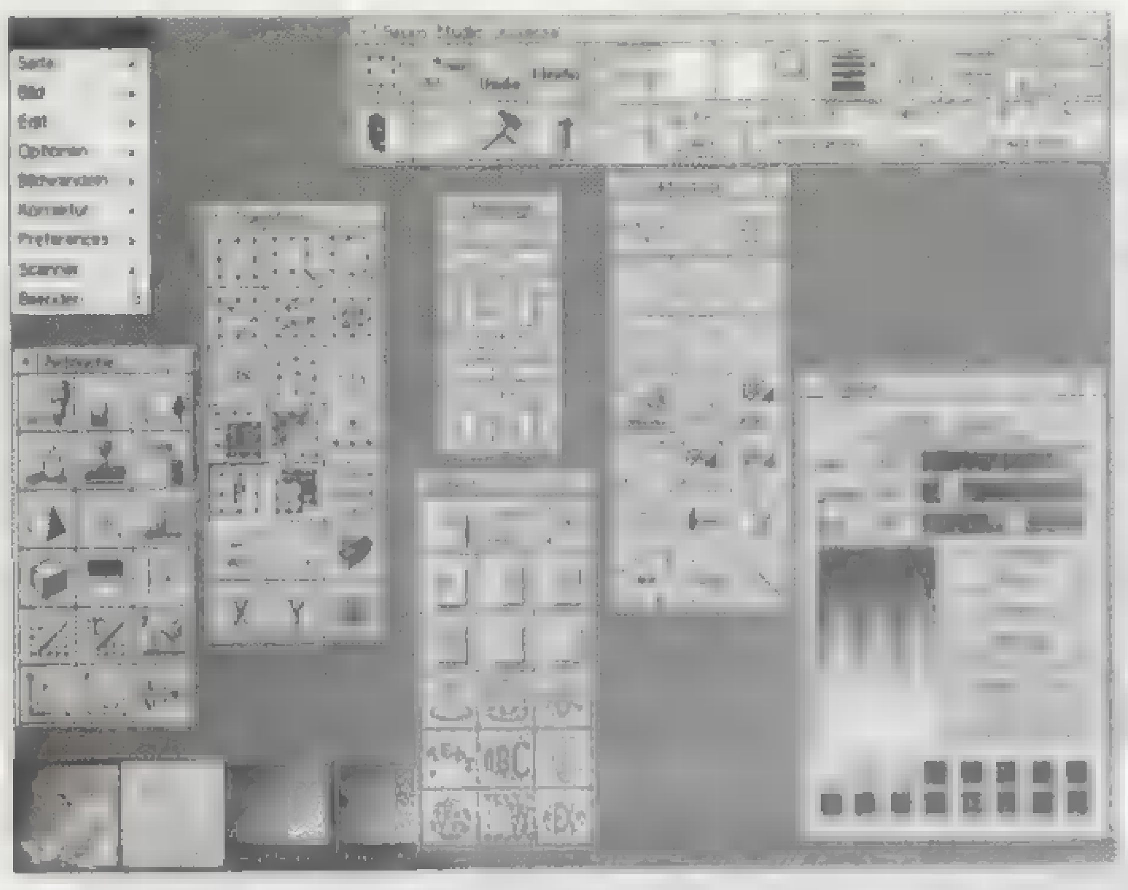
Próbę wypełnienia tej luki podjęła niemiecka firma Palatinum GmbH wypuszczając pakiet o nazwie "Repro Studio Universal". Jest to pierwszy produkt pracujący w systemie okienkowym, a nie ekranowym, co pozwala na bezproblemową pracę z różnymi kartami graficznymi. Dotychczasowe pakiety otwierając wiele różnych ekranów nie dawały takiej sposobności.

Amigowe karty graficzne nie umożliwiają otwierania przesuwających się między sobą ekranów (Wyjątek stanowi Retina, której oprogramowanie daje taką możliwość. Jednak otwarte ekrany muszą być o takich samych parametrach.) Niedogodność ta wiąże się z ilością pamięci karty. Zauważmy, że jeden ekran w rozdzielczości 800 na 600 punktów i głębokości 24 bity zajmuje 1,37 MB. W związku z tym, już dwa otwarte ekrany wymagają minimum 2,74 MB pamięci karty. Z tego wynika, że nie mogą być one otwarte jednocześnie (maksymalna pamięć kart graficznych to 2 MB, Retina - 4 MB), więc oprogramowanie ładuje aktywny ekran do swojej pamięci, a pozostałe przechowuje w pamięci CHIP lub FAST komputera (w zależności od karty).

Aby zapewnić bezproblemową pracę w 24 bitach, w Repro Studio Universal odstąpiono od obsługi okienek poprzez standardowe biblioteki amigowskie i stworzono własny system o nazwie "AWindows". Decyzja ta została spowodowana tym, że biblioteki systemowe potrafią otworzyć okna w maksymalnie 8 bitach, ewentualnie w HAMie 8. AWindows pracuje wewnętrznie zawsze w 24 bitach i potrafi obsłużyć oprócz trybów amigowych (do 8 bitów) inne karty graficzne, takie jak Retina, Picasso II oraz karty systemu EGS (EGS 110/24, EGS 28/24, Piccolo, Rainbow III). Ilość zastosowanych kolorów jest ograniczona tylko możliwościami karty. Bardzo wygodna jest praca w wysokiej rozdzielczości np. 1024*768 i 65536 kolorów. Wadą zastosowanego systemu jest to, że w trybach amigowskich pracuje niesamowicie wolno, do tego stopnia, że na A4000/040 w trybie 8 bitów praca jest właściwie niemożliwa i koniecznie trzeba zastosować jakąkolwiek kartę graficzną. Do systemu AWindows autorzy dołączyli kilka pomocnych programików m. in. zegarek, wygaszacz ekranu, preferencje itp.

Przejdźmy teraz do programu podstawowego. Repro Studio Universal otwiera specjalne okno, z którego mamy dostęp do prawie wszystkich opcji. Tak więc możemy wywołać pięć okienek: cztery narzędziowe oraz okno palety. Każde z nich odpowiada innej grupie czynności. I tak pierwsze oferuje narzędzia malarskie i retuszerskie, drugie dotyczy pracy na tekście, trzecie służy do tworzenia grafiki wektorowej,





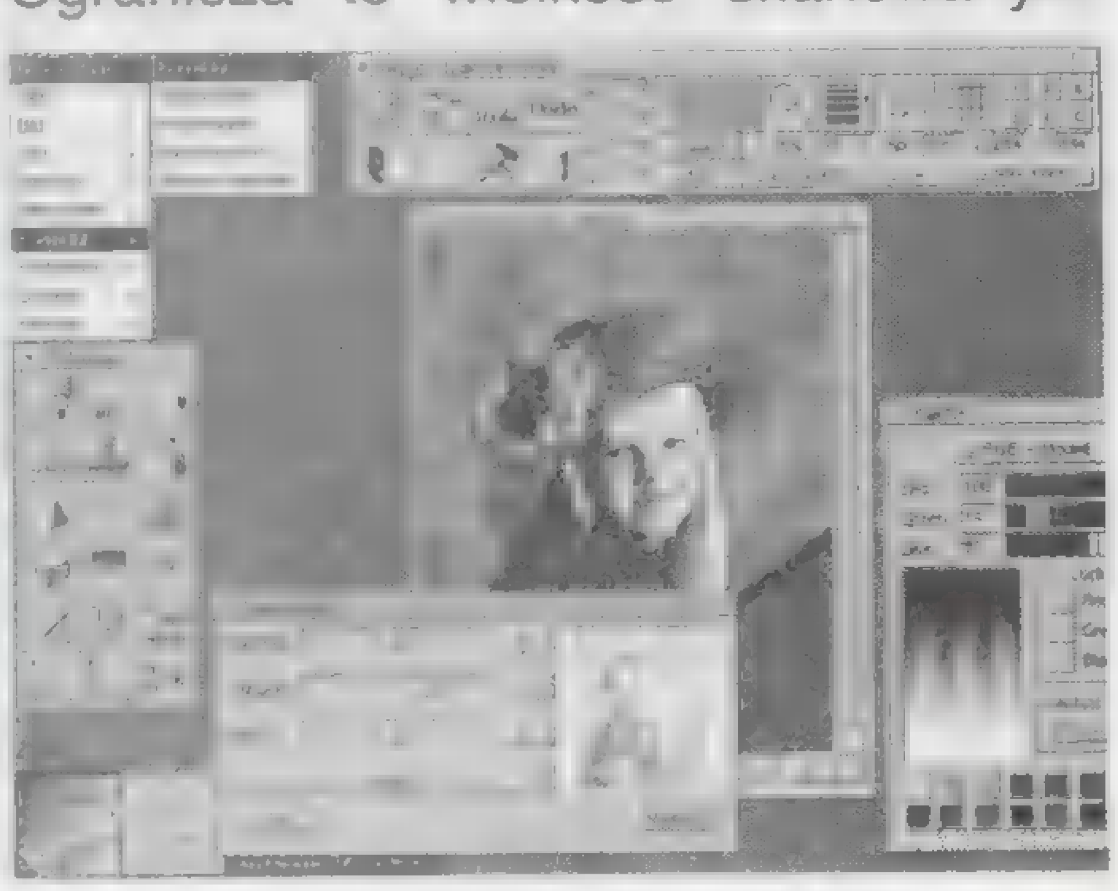


czwarte do zarządzania obiektami, transformacji obiektów i precyzyjnego umieszczania obiektów na obrazku. Niektórzy mogą dziwić się, dlaczego istnieją narzędzia do grafiki wektorowej w programie do obróbki grafiki rastrowej. Otóż Repro Studio traktuje grafikę w dosyć ciekawy sposób, a mianowicie np. wycięty kawałek obrazka (popularny brush) stanowi osobny element, który można przesuwać, zniekształcać, nadawać przezroczystość, itp. Leży on jakby na warstwie podstawowej obrazu. Największą rewelację stanowi jednak to, że możemy stworzyć następny wycinek i jeszcze następny, a wszystkie są do naszej dyspozycji. Możemy zmieniać kolejność ich ułożenia, jeden na spód, inny na wierzch. Można nawet założyć taką sytuację, że podstawowym obrazem jest pusta biała kartka, a cała ilustracja składa się z wolnych elementów. Ponadto fakt, że każda operacja wykonywana na danym obiekcie będzie cały czas widoczna np. przezroczystość, maska czyni taki system zarządzania bardzo wygodnym. Program traktuje elementy obrazu bitowego tak, jakby były to elementy wektorowe. Jeśli więc wybierzemy narzędzie rysowania prostokąta, które jest dostępne w bloku grafiki wektorowej, to po narysowaniu będziemy mieli możliwość jego poprawienia, a także wsunięcia pod jakiś istniejący już element bitowy. Oczywiście należy pamiętać, że po zgraniu takiej ilustracji w formacie IFF, wszystkie obiekty zostaną ze sobą połączone i zamienione na grafikę rastrową, czyli uzyskamy to, o co nam chodzi. W trakcie pracy, aby nie tracić elementów można zapamiętać tworzony projekt w specjalnym formacie Repro Studio. Same elementy grafiki wektorowej można również zapisać w osobnym pliku i wczytać kiedy będą potrzebne. Dzięki takiemu rozwiązaniu systemu pracy, programem tym możemy tworzyć zarówno obrazki bitowe jak i wektorowe. Do mankamentów można zaliczyć fakt, że wektorówka może być zapisana tylko w formacie CGM, który w środowisku Amigi nie jest w ogóle znany. Na szczęście producent już w instrukcji obsługi zapewnia, że niedługo będą istniały inne moduły eksportu.

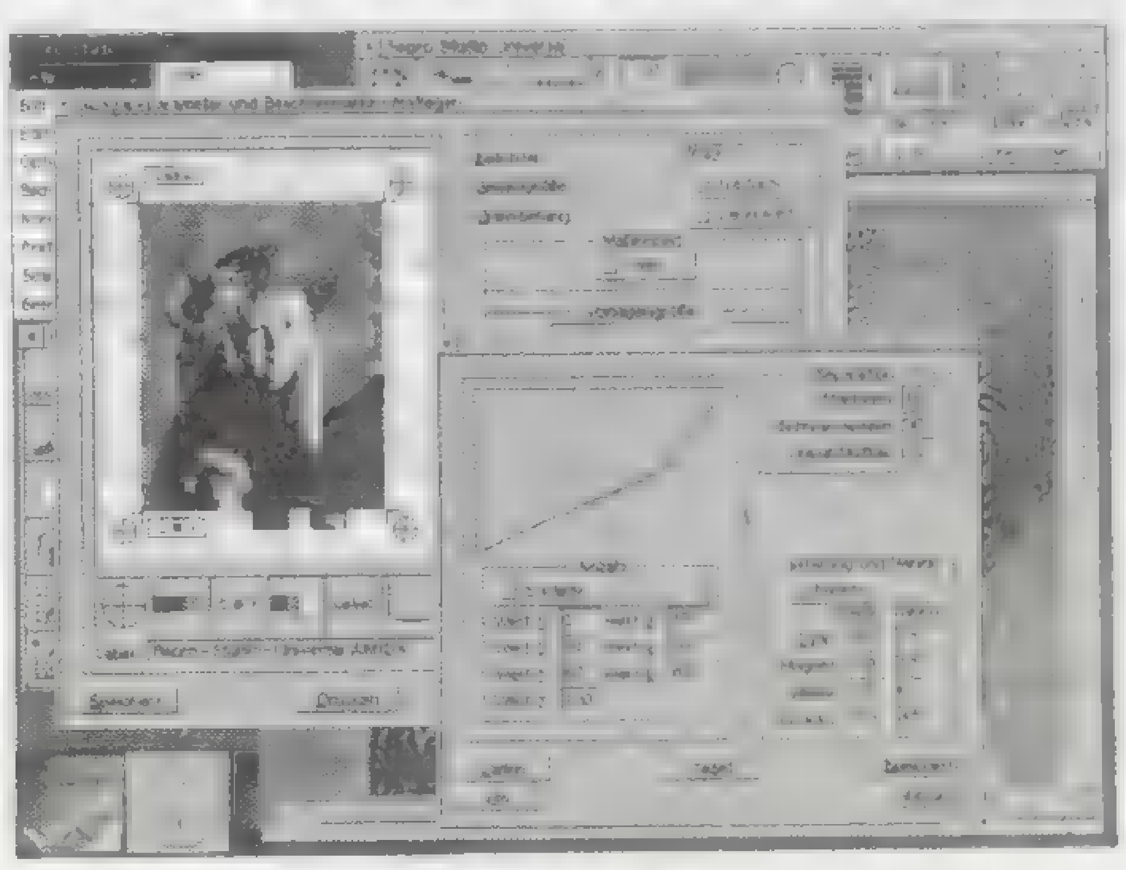
Omówmy teraz narzędzia retuszerskie. Istnieją oczywiście takie podstawowe jak pędzelek, mazak czy spray, lecz ich moc odkrywamy dopiero, gdy przy ich wywołaniu naciśniemy drugi klawisz myszy. Otwiera się wtedy specjalny requester, w którym możemy ustawić wiele opcji. Z innych narzędzi tej grupy należałoby wymienić palec do rozmazywania, farbkę wodną, nożyk do wyostrzania krawędzi, gąbkę do ich zmiękczania, wiaderko do wypełnia nia, dostęp do różnorakich filtrów, itp.

Bardzo rozbudowane są narzędzia do obróbki tekstu. Nadmienić trzeba, że tu również zastosowano tryb pracy opartej na wektorach. Program wykorzystuje czcionki wektorowe Adobe Type 1. Po napisaniu tekstu oprócz nadania mu koloru możemy go dowolnie formatować, równać do dowolnej krawędzi. Możliwe są także transformacje geometryczne takie jak nakładanie na kulę czy na walec.

Repro Studio Universal umożliwia oczywiście korzystanie z kolorowego skanera. W tej chwili posiada drivery do następujących typów: Epson, Mustek, Arti Scan, Tamarack. Pewną wadą istniejących obecnie sterowników jest fakt, że skanowanie odbywa się do pamięci RAM, a nie na twardy dysk. Ogranicza to wielkość skanowanych ilustracji (czasami trzeba skanować w dużej rozdzielczości, a taki skan może zajmować i powyżej 20 MB). Producent programu poinformował nas, że trwają prace nad nowymi sterownikami, które całkowicie rozwiążą ten problem. Skoro już mowa o skanowaniu, to często ważniejsza od rysowania jest możliwość nadawania pewnych korekcji obrazom uzyskanym tym sposobem. Dlatego też Repro Studio taką możliwość posiada. Rozwiązanie tej opcji jest wygodniejsze w innych amigowych programach. Jeśli chcemy np. zwiększyć nasycenie koloru czerwonego i zmniejszyć kontrast ilustracji, wybieramy opcję z menu. W wyniku tego otwiera się specjalne okno, które oprócz suwaków odpowiedzialnych za korekcję posiada wydzielony obszar, w którym widzimy pomniejszony obrazek, nad którym pracujemy. Wszystkie wprowadzane korekcje widzimy natychmiastowo właśnie na tej przeglądowej ilustracji.



Na zakończenie wspomnę jeszcze, że program umożliwia wykonanie prawidłowych separacji barwnych również jako plik postscriptowy EPS z przeglądówką. Przy separacji mamy możliwość zmian takich parametrów jak kompensacja czerni, UCR, GCR, gęstość i kąt rastra każdej ze składowej barwy oraz wybór jednego z 22 typów rastrowania (od punktowego przez liniowe, kołowe, aż do specjalnych wzorków). Jednym słowem program wygląda bardzo pozytywnie. Niestety wkradło się parę błędów w jego działaniu, które w obecnej wersji powodują, że praca nie jest pewna i może nabawić wielu stresów. Jako że jest to dopiero pierwsza wersja programu producent zapewnia, że błędy zostaną poprawione. Na pocieszenie należy dodać, że taki osławiony program jak Corel Draw w swojej wersji czwartej posiada tyle błędów, że musiano wypuścić do niego aż siedem dysków z poprawkami. Widocznie firmy programistyczne stwierdziły, że taniej jest testować program wśród użytkowników niż w firmie. ❑



IMAGINE
Kurs cz.1

Marek Stor

Po długich namowach dałem się złamać. Tak, napiszę coś o Imagine. Ale nie liczcie na to, że będzie to instrukcja do programu. Nie chcę wyręczać firmy Impulse w pisaniu instrukcji, która krótko mówiąc do wersji 1.0 jest beznadziejna, ale już do wersji 2.0 stoi na przyzwoitym poziomie. Kto z Was nie zna jednego z najpopularniejszych programów na Amigę (a teraz również na PeCeta) służącego do tworzenia animacji trójwymiarowej?

Chcę podzielić się z wami kilkoma sztuczkami, które czynią pracę na Imaginie przyjemniejszą i łatwiejszą. Na wiele z tych tricków nie wpadłem sam, ale chyba nikt nie jest w stanie odkryć wszystkiego. Sam wielu rzeczy jeszcze nie wiem i z niecierpliwością będę czekał na pomoc ze strony czytelników. Postaram się pod każdym przykładem podać źródło, z którego zaczerpnąłem pomysł a także, aby wszystkie przykłady zostały zamieszczone na dyskietkach PD Amigowca. Żeby nie było wątpliwości, posiadam Amigę 3000 z 18MB RAMu - twardym 480MB i raczej nie mam problemów z generowaniem animacji, poza jednym: ZA WOLNO! Wszystkie przykłady są przeze mnie sprawdzone na Imagine v.2.0.

Każdą z opcji będę opisywał krok po kroku, aby uniknąć niepotrzebnych nerwów z Waszej strony, ponieważ to co dla mnie jest proste dla innych może być niezrozumiałe.

Zaczynamy.

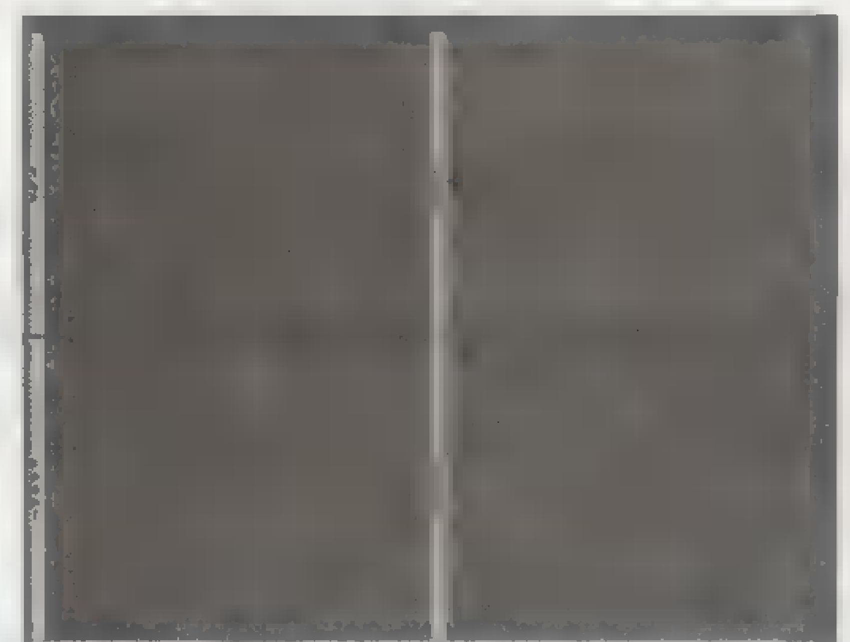
Propozycja 1

Na początku polecam wam zmodyfikowanie listwy w Imagine. Po wielu próbach i podpatrywaniu jak to robią inni doszedłem do następującego ustawienia. Ustawienie takie pozwoli wam ograniczyć jeżdżenie myszą po ekranie i otwieranie górnych menu.

Krok.1 Otwórz Imagine'a. Wybierz z górnej listwy PREFERENCES # User Gadgets

(Patrz Rysunek 1)
Zmiany dla DETAIL EDITOR.

```
DBTN  210,"Kord"      # Coordinates on/off
DBTN  260,"ZI"        # Zoom in
DBTN  270,"ZO"        # Zoom out
DBTN  4d1,"Mag"       # Magnetism on/off
DBTN  4d2,"MS"        # Magnetism Setup
DBTN  584,"Prim"      # Add primative
DBTN  6a0,"Find"      # Find requester
```


Rysunek 1.


Rysunek 2.

Dopisz

```
DBTN  2c0,"Sh"        # Solid Preset
DBTN  390,"Attr"      # Attributes
DBTN  4e2,"Box"       # Drag Box
DBTN  4e3,"Las"       # Lasso
DBTN  4e1,"Cl"        # Klick
DBTN  220,"I"         # Interlace
```

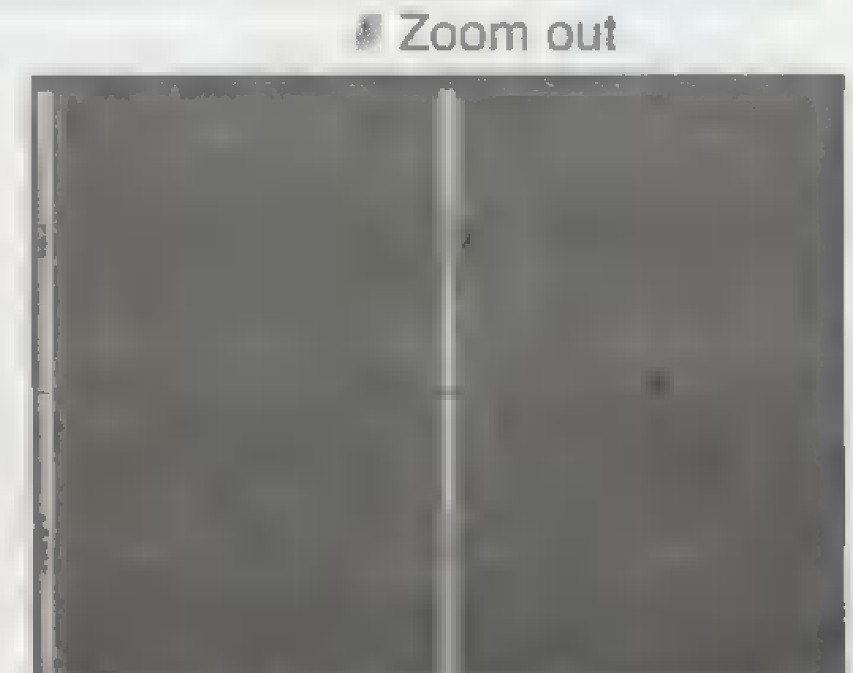
(Patrz Rysunek 2)
Zmiany dla FORMS EDITOR

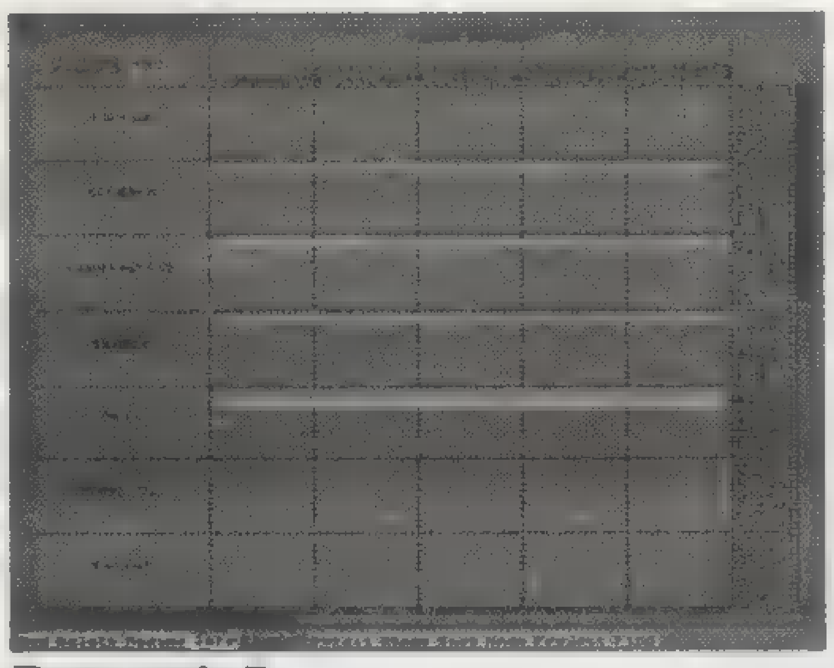
```
FBTN  210,"Coords"    # Coordinates On/Off
FBTN  260,"ZI"        # Zoom in
FBTN  270,"ZO"        # Zoom out
FBTN  510,"...NoSym"  # Symmetry off
FBTN  520,"FS"        # Front (Top) view symmetry
FBTN  530,"RS/TS"     # Right (Front) view symmetry
FBTN  540,"Both"      # Symmetry in both views
FBTN  550,"90"        # "90 degree" symmetry
FBTN  710,"...Select" # Select cross section
FBTN  720,"+Key"      # Make key cross section
FBTN  730,"-Key"      # Unmake key cross section
FBTN  740,"Copy"      # Copy from another cross section
FBTN  750,"Cancel"    # Cancel cross section command
FBTN  410,"...Edit"   # Edit mode
FBTN  420,"Add"       # Add mode
FBTN  430,"Del"       # Delete mode Dopisz
FBTN  170,"Un"        # Undo
```

(Patrz Rysunek 3)
Zmiany dla CYCLE EDITOR.

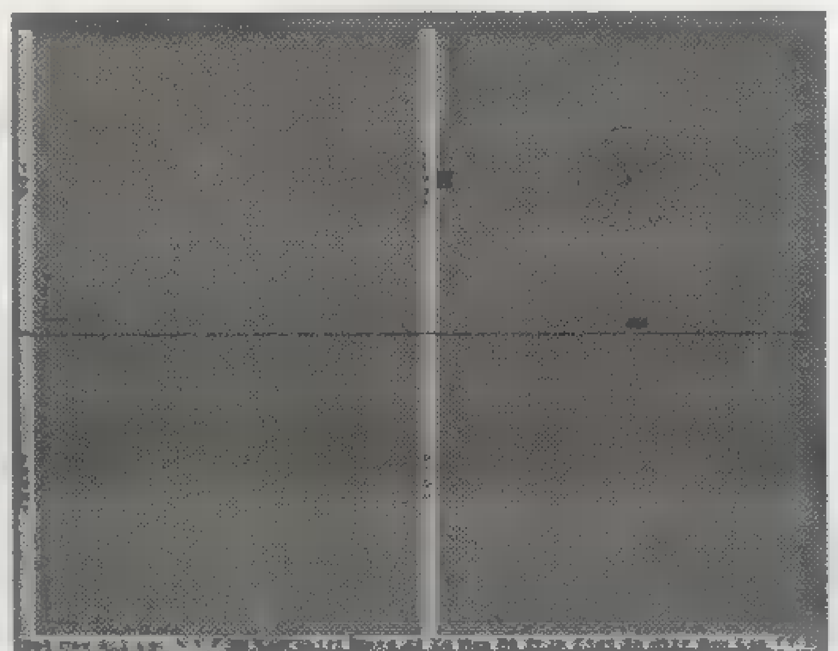
```
CBTN  210,"Coords"    # Coordinates on/off
CBTN  260,"ZI"        # Zoom in
CBTN  270,"ZO"        # Zoom out
```


Rysunek 3.


Rysunek 4.



Rysunek 5.



Rysunek 6.

Dopisz

```
CBTN    2c0,"Shade"     # Shadet
CBTN    2a0,"Wire"      # Wireframe
CBTN    120,"Detail"    # Detail Editor
CBTN    6d0,"LPose"     # Load Pose
CBTN    650,"Goto"      # Goto
CBTN    610,"First"     # First Cell
CBTN    710,"A-Make"    # Make Anim
CBTN    740,"P-Loop"    # Play Loop
CBTN    170,"Undo"      # Undo
CBTN    330,"Save"
```

(Patrz Rysunek 4)
Zmiany dla STAGE EDITOR.

```
SBTN    210,"Cord"      # Coordinates on/off
SBTN    260,"ZI"        # Zoom in
SBTN    270,"ZO"        # Zoom out
```

Dopisz

```
SBTN    322,"Ligh"      # Add Lichtquelle
SBTN    2d0,"CamV"      # Camera View
SBTN    750,"Got"       # Goto
SBTN    730,"Nex"       # Next Frame
SBTN    3b0,"Trc"       # Camera Retrack
SBTN    310,"Load"      # Load Object
SBTN    2c0,"Sh"        # Shade Preview
SBTN    160,"S"         # Save Changes
SBTN    220,"I"         # Interlace
SBTN    150,"A"         # Actions Editor
SBTN    110,"P"         # Project Editor
```

(Patrz Rysunek 5)
Zmiany dla ACTION EDITOR.

```
ABTN    210,"Add"       # Add action bar mode
ABTN    220,"Delete"    # Delete action bar mode
ABTN    230,"Rename"    # Rename stage object mode
ABTN    240,"Info"      # Action bar info mode
ABTN    270,"Cancel"    # Cancel Add
```

Dopisz

```
ABTN    120,"Detail"    # Detail Editor
ABTN    150,"Stage"     # Stage Editor
ABTN    160,"Save"      # Save Changes
ABTN    180,"Undo"      # Undo
ABTN    110,"Project"   # Project Editor
ABTN    250,"Find"      # Find
```

Takie ustawienie konfiguracji jest tylko propozycją, zrobicie tak jak będzie Wam pasować. Chętnie usłyszę o Waszych pomysłach.

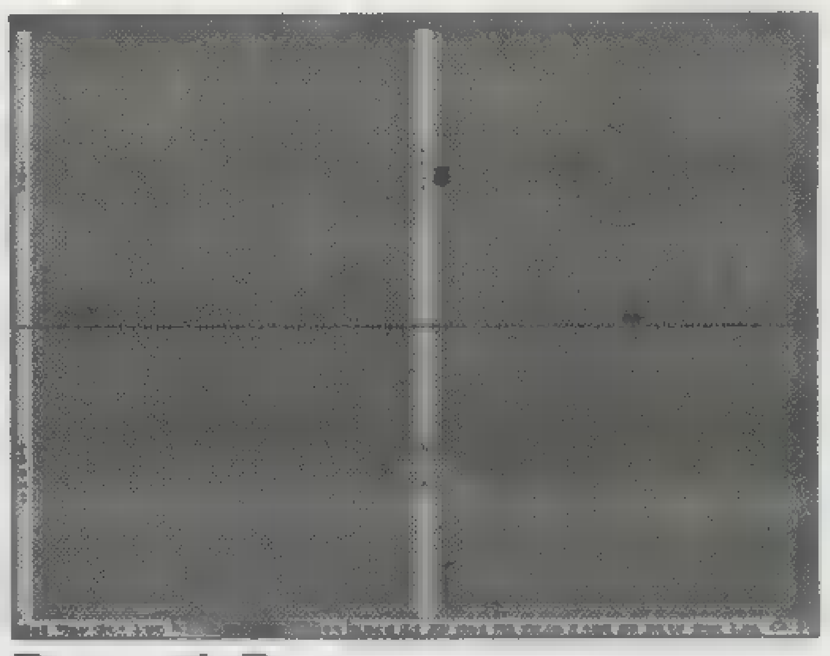
Mamy już program dostosowany do pracy, ■ teraz zajmijmy się czymś praktyczniejszym. Czy zastanawialiście się jak zrobić spiralę na Imagine? Wbrew pozorom jest to bardzo proste. Pomysł został zaczerpnięty z książki "Das Imagine Buch" pana Marco Vitolini-Naldini, ■ przeze mnie tylko trochę rozbudowany.

Krok I

1. Wejdź w DETAL EDITOR.

Z FUNCTION Menu wybierz opcję tworzenia brył (ADD/PRIMITIVES) lub naciśnij F5. Najedź strzałką na requester DISK i naciśnij lewy przycisk myszy. Wpisz teraz parametry obiektu jaki chcesz uzyskać np.
RADIUS : 40
SECTIONS : 12
i naciśnij OK. *(Patrz Rysunek 6)*

2. Kliknij teraz na środek stworzonego przez siebie obiektu, tak aby zmienił kolor na fioletowy lub naciśnij F1. Następnie z MODE Menu wybierz opcję PICK-POINT. Zostaną pokazane wszystkie punkty, z których składa się obiekt. Kliknij myszą na środkowy punkt tak, aby zmienił kolor i wybierz DELETE z FUNKCTION Menu (lub naciśnij prawy klawisz Amiga i D). Wybierz teraz funkcję PICK-OBJECT z MODE Menu, aby twój obiekt uzyskał kolor fioletowy. Uzyskałeś teraz obiekt składający się z samej obręczy. *(Patrz Rysunek 7)*



Rysunek 7.

3. Następnym etapem jest wyznaczenie środka naszej spirali. Wybierz z OBJECT Menu funkcję TRANSFORMATION (prawa Amiga + T), aby przesunąć oś obiektu o 90 stopni względem osi X. Kliknij myszką na pole o nazwie ROTATE i wpisz dla X wartość 90. Następnie kliknij na polu TRANSFORM AXIS ONLY (chcemy obrócić tylko oś obiektu, ■ nie cały obiekt). Teraz kliknij na PERFORM, który zatwierdza dokonane zmiany.

4. Ustalmy teraz promień jaki ma mieć nasza spirala. Przy pomocy Imagine robi się to w dwóch krokach. Wywołaj jeszcze raz funkcję TRANSFORMATION i przesuń obiekt o sto jednostek w lewo. Kliknij na polu TRANSLATE i wpisz dla X = -100 i naciśnij PERFORM. Teraz oś obiektu musi wrócić na stare miejsce. Wywołaj TRANSFORMATION i przesuń oś obiektu o sto jednostek w prawo (TRANSLATE X = 100; TRANSLATE AXIS ONLY, PERFORM). Twój obiekt jest gotowy do rozciągania. *(Patrz Rysunek 8)*

5. Wybierz z OBJECT Menu funkcję MOLD, ■ potem funkcję EXTRUDE. Teraz musimy ustalić wysokość naszej spirali oraz z ilu pierścieni ma być zbudowana. Załóżmy, że spirala będzie się składać z trzech pierścieni. Podczas jednego obrotu spirala będzie rosnąć o 40 jednostek tak, że trzy spirale będą miały wysokość 120 jednostek. (3 x 40). Dla parametru LENGHT wpisz 200 jednostek (nie 120, ponieważ między spiralami muszą być jakieś odstępy).

6. Teraz musimy wyznaczyć ilość sekcji z jakich będzie składał się nasz obiekt podczas wytłaczania. Ponieważ chcemy obrócić spiralę trzy razy wokół osi, niezbędne jest ustalenie liczby duplikatów. Ustalamy 24 sekcje na jeden obrót spirali, więc na trzy obroty wychodzi 72 sekcje.

7. Ostatnim parametrem jest Y-ROTATION, w którym ustalamy o ile stopni obróci się spirala względem osi Y. Dla naszych trzech obrotów wpisujemy wartość 1080 (jest to 360 stopni razy trzy obroty).

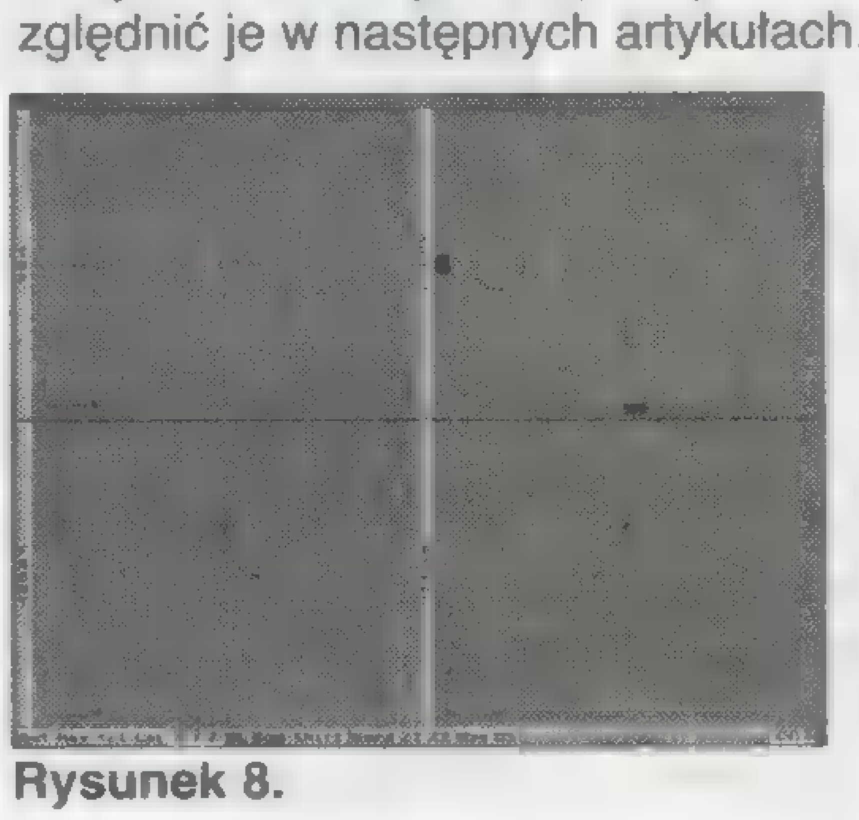
Powinieneś mieć wpisane takie parametry:

LENGHT		:200
SECTION		:72
Y-ROTATION		:1080

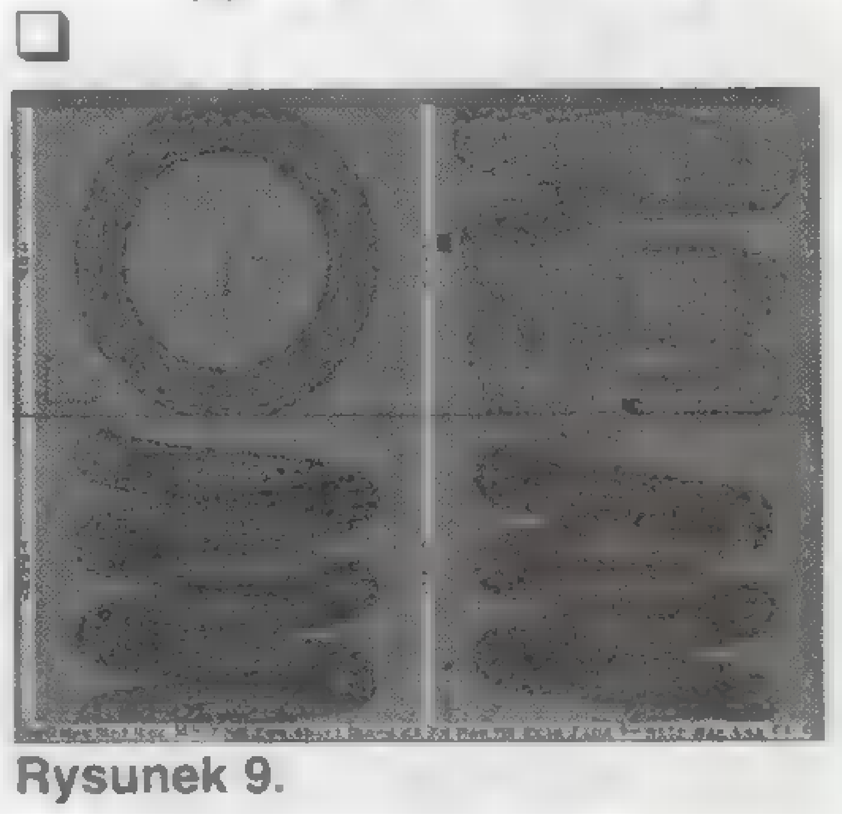
i kliknij na OK. *(Patrz Rysunek 9)*

8. Podziwiaj to co stworzyłeś. Jeżeli wyszło Ci coś innego niż na rysunku poniżej, to spróbuj jeszcze raz od początku. Proponuję także poeksperymentować z parametrami wytłaczania.

Jest to pierwsze nasze spotkanie. Myślę, że przy następnym dotrą do mnie Wasze sugestie, gdyż chciałbym abyście byli twórcami tej rubryki. Piszcie jakie sprawy Was interesują, a ja postaram się uwzględnić je w następnych artykułach. ❑



Rysunek 8.



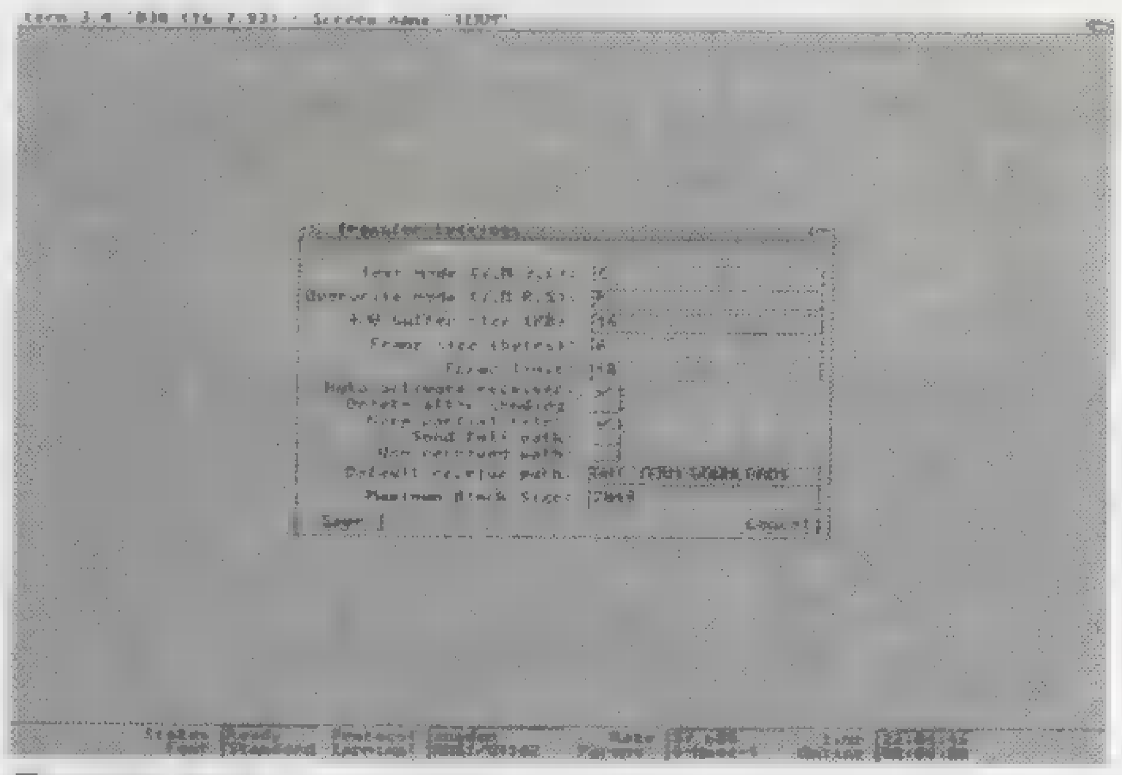
Rysunek 9.

Dlaczego? TERM

Tomasz Matajek

W poprzednim odcinku pokazane zostało jak w prosty i szybki sposób można rozpocząć pracę z terminalem. Dzisiaj przyjrzymy się dokładniej niektórym punktom z menu.

Zacznijmy od protokołu transmisji ZModem. Ma on kilka ciekawych opcji, znacznie ułatwiających życie. Aby się do nich dostać wybieramy punkt "Transfer protocol options" z menu "Settings". Patrz Rysunek 1.



Rysunek 1.

Jak widać jest co ustawiać. Jednak co oznaczają kolejne parametry? Zacznijmy od początku:

Text mode (Y, N, ?, C)

- parametr "Y" powoduje konwersję odbieranych plików tekstowych w ten sposób, że kody CR+LF końca i nowej linii (używane w komputerach IBM) lub tylko CR (MacIntosh), zamieniane są na kod LF używany w Amidze. Przy wysyłaniu pliku tekstowego, program odbierający informowany jest, że przesyłany plik powinien być przyjmowany w trybie tekstowym.

- parametr "N" oznacza brak konwersji. Przesyłane dane odbierane są w niezmienionej postaci.

- parametr "?" znaczy, że podczas odbioru, tekst będzie konwertowany tylko na żądanie programu nadającego. Nadawany plik tekstowy będzie natomiast po drugiej stronie odebrany w standardowym dla odbiornika modzie (ustawieniu).

- parametr "C" działa tak, że nadawany lub odbierany tekst będzie podlegał specjalnej konwersji ustalonej przez program komunikacyjny.

Overwrite mode (Y, N, R, S)

- Gdy ustawimy "Y", to odbierany plik spowoduje skasowanie ewentualnie już istniejącego pliku o tej samej nazwie.

- "N" (overwrite mode = no). Ewentualnie istniejący plik o tej samej nazwie co aktualnie odbierany nie zostanie skasowany, a nazwa nowego zostanie rozszerzona o końcówkę ".dup" (duplikat).

- Ustawienie "R" (resume - kontynuacja) to chyba najciekawsza możliwość. Jeśli z jakichś powodów nie udało się odebrać danego pliku w całości, to nie trzeba ponownie odbierać go od początku, lecz dokładnie od miejsca, w którym transmisja została przerwana. Ma to olbrzymie znaczenie przy przesyłaniu długich plików - bez takiej możliwości ich transmisja nie byłaby czasami w ogóle możliwa. Polecam wybranie właśnie tej opcji.

- "S" (skip) spowoduje zaniechanie transmisji, jeśli plik o danej nazwie już istnieje. Przesyłanie pozostałych plików będzie kontynuowane.

Buffer size

- ustawia dodatkowy bufor transmisji niezależnie od bufora ustalonego przez program. Minimalna wielkość to 1 KB, maksymylna jest ograniczona wielkością dostępnej pamięci. Praktyczna wartość to 16 KB (dla niezbyt szybkich twardych dysków) lub mniejsza (dla szybkich dysków lub RAM-dysku).

Frame Size

- normalnie opcja ta powinna być wyłączona (wartość ustawiona na 0). Wpisanie wartości n większej niż zero wprowadza żądanie dodatkowego potwierdzenia przesłania n-bajtowego pakietu danych. Znacznie spowalnia to transmisję i jest przydatne tylko w szczególnych przypadkach (np. praca ze stacją dysków w "przeciążonym" multitaskingu).

Error limit

- określa ilość kolejnych błędów wymaganą przez protokół do przerwania transmisji. Wielkość 10 jest odpowiednia nawet dla fatalnych krajowych centrali telefonicznych.

Auto-activate receiver

- powoduje automatyczne przełączenie terminala w tryb odbioru po wykryciu kodu rozpoczęcia transmisji. Jest to przydatna opcja ułatwiająca pracę.

Delete after sending

- UWAGA: gdy opcja ta jest aktywna, to po transmisji przesyłane pliki zostaną skasowane.

Keep partial files

- zaktywizowanie tego gadżetu spowoduje, że niekompletne pliki nie zostaną skasowane nawet wtedy, gdy przerwanie transmisji nastąpiło na żądanie użytkownika. Dzięki temu możliwe będzie wykorzystanie omówionej wcześniej opcji "overwrite resume".

Send full path

- po ustawieniu tego parametru przesyłane będą nazwy plików wraz z kompletną ścieżką dostępu, np. "dh0:tols /commodities/blanker", w przeciwnym razie przesłana zostanie jedynie nazwa główna.

Use received path

- wybranie tej opcji spowoduje wykorzystanie oryginalnej ścieżki odbieranego pliku.

Default directory path

- tutaj należy wpisać nazwę szuflady, w której mają być umieszczane odbierane pliki. Działa to oczywiście tylko wtedy, gdy opcja "Use received path" opisana wyżej, nie jest aktywna.

Maximum block size

- protokół ZModem umożliwia przesyłanie danych w blokach o wielkościach 64, 128, 256, 512, 1024, 2048, 4096 i 8192 bajty. Standardowo ustala się tę wartość na 1024, ale o najkorzystniejszym ustawieniu decyduje kilka czynników. Są to:
- jakość linii telefonicznej,
- szybkość połączenia,
- szybkość komputera.

Przy bardzo marnej jakości linii telefonicznej użyteczne są małe pakiety (poniżej 1024), gdyż przy wystąpieniu błędu retransmitowany będzie stosunkowo krótki blok. Stosowanie dużych bloków (ponad 2048) ma sens wtedy, gdy połączenie odbywa się na prędkości powyżej 9600 bodów, gdyż czas pomiędzy dwoma kolejnymi blokami staje się stosunkowo krótki. Ma to tym większe znaczenie, im wolniejszy mamy komputer, ponieważ poprawność każdego odebranego bloku danych musi zostać sprawdzona przez procesor. Nie należy jednak ustawiać wartości mniejszych niż 1024, dlatego, że po pojawieniu się błędów transmisji protokół ZModem automatycznie zmiejsza używaną długość bloku, nawet do minimalnej wartości 64 bajty.

Kolejnym ważnym punktem z menu "Settings" jest "Modem". Requester wygląda tak jak na Rysunku 2.

Modem init command

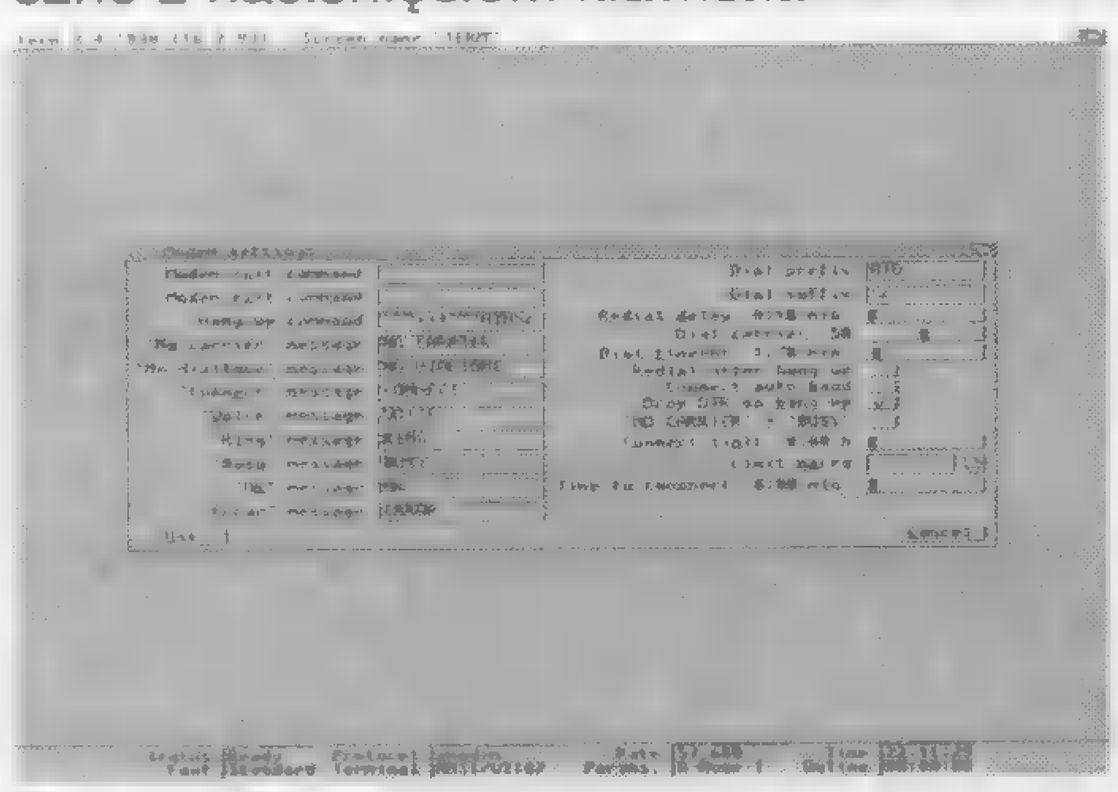
- można tu wpisać komendę, która będzie wysyłana do modemu zaraz po uruchomieniu programu, oraz każdorazowo przed wybraniem danego numeru telefonicznego. Normalnie miejsce to zostawia się wolne.

Modem exit command

- funkcja bardzo podobna do opisanej wyżej. Można tu wpisać komendę, która będzie wysyłana do modemu tuż przed wyjściem z programu, a także każdorazowo po wybraniu danego numeru telefonicznego. Normalnie miejsce to zostawia się wolne.

Hang up command

- wpisuje się tu komendę powodującą "odwieszenie słuchawki", czyli zwykłe rozłączenie. Praktycznie dla wszystkich modemów jest to sekwencja ~~~+++~~~ATHr. Co oznaczają te tajemnicze znaczki? Pojdynczy znak ~ oznacza pół sekundową przerwę. Trzy plusy wymuszają przejście modemu do stanu, w którym przyjmuje komendy. Przed i po sekwencji trzech plusów wymagana jest co najmniej sekundowa przerwa, dlatego dla bezpieczeństwa wykorzystane są trzy znaki ~. ATH jest właściwą komendą kasującą połączenie, a r jest równoznaczne z naciśnięciem klawisza RETURN.



Rysunek 2.

Wszystkie "messages" (informacje) mówią same za siebie. Można tu wpisać teksty, które pojawią się na ekranie jako reakcja na odpowiednie zdarzenia. Np. gdy modem gdzieś dzwoni, a linia jest zajęta, na ekranie pojawi się komunikat "BUSY". Zamiast tego można oczywiście wpisać cokolwiek, chociażby "Niestety, linia jest zajęta!"

Dial prefix

- podaje się tu string (ciąg znaków), który powoduje rozpoczęcie dzwonienia. Przeważnie jest to ATD.

Dial suffix

- wpisuje się tutaj string, który będzie dodany bezpośrednio po komendzie ustalonej przez "dial prefix". Zwykle jest to r, czyli RETURN.

Redial delay

- ustala długość przerwy pomiędzy kolejnymi próbami dodzwonienia się pod dany numer.

Dial retries

- ustala maksymalną liczbę prób dodzwonienia się pod dany numer.

Dial timeout

- określa maksymalny czas oczekiwania na połączenie podczas wykonywania procedury dzwonienia. Zwykle ustala się na 1 - 2 min.

Redial after hang up

- terminal może próbować dzwonić kolejno pod kilka numerów. Jeżeli aktualne połączenie zostało przerwane lub normalnie zakończone, spróbuje on zadzwonić pod następny numer z listy.

Connect auto-baud

- cel tej opcji nie jest do końca jasny, ogólnie rzecz biorąc nie należy jej używać.

Drop DTR on hang up

- niektóre modemy śledzą zachowanie linii DTR. Zmiana jej stanu spowoduje rozłączenie.

No carrier = busy

- gdy opcja ta jest włączona, reakcja modemu na brak połączenia będzie taka sama jak na sygnał zajętości.

Connect limit

- jest to opcja używana dla własnego bezpieczeństwa. Ustala ona maksymalny czas połączenia, aby nie zbankrutować na rachunkach telefonicznych.

Time to connect

- ustala maksymalny czas pomiędzy "podniesieniem słuchawki" przez drugi modem, a uzyskaniem efektywnego połączenia. Jeśli po upływie tego czasu modemy nie "dogadają się", połączenie zostanie przerwane.

Mam nadzieję, że opisane wyżej opcje umożliwią każdemu skonfigurowanie terminala stosownie do indywidualnych potrzeb. Bardziej dociekliwym zdecydowanie polecam oryginalny opis załączony wraz z programem. ❑

TEORIA DŹWIĘKU

Tomasz Flaar

Obcując na co dzień z użytkownikami różnych komputerów, a zwłaszcza Amigi, można spotkać się z takim pojęciem jak SAMPLING. Nazwa ta oznacza specjalną technikę obróbki dźwięku, która polega na przetwarzaniu sygnału na postać cyfrową.

Wszyscy, którzy zajmowali się kiedykolwiek samplowaniem na Amidze wiedzą, że istnieje kilka standardów składowania uzyskanych tą drogą danych. Na froncie 8-bitowym użytkownicy Amigi mają do dyspozycji chyba z pół tuzina rozmaitych formatów zapisu. Ostatnio producenci oprogramowania służącego do ściągania i obróbki sampli skłaniają się do faworyzowania dwóch standardów: IFF 8SVX i RAW. Wybór jest więc prosty, aby jednak dokonać go świadomie, należy wiedzieć coś więcej na temat wspomnianych standardów. (Patrz Rysunek 1).



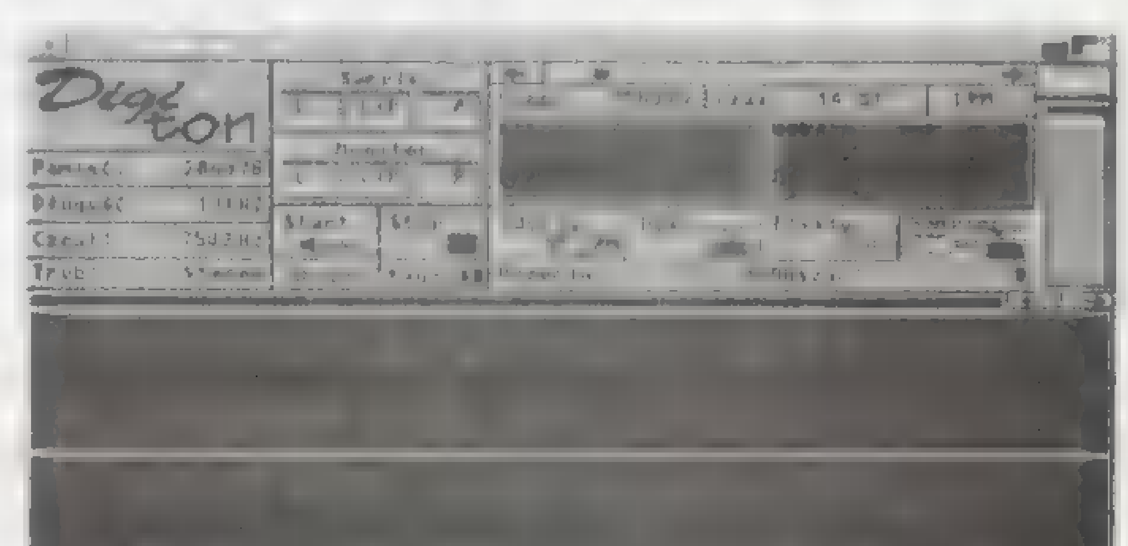
Rysunek 1.

Prawie każdy Amigowiec musiał słyszeć o standardzie IFF (Interchangeable File Format - wymienialnym formacie plików). Jest to powszechnie przyjęty standard składowania wszelakiego rodzaju danych, wprowadzony prawie we wszystkich programach, w których używane są operacje zapisu i odczytu. Zamierzeniem twórców tego standardu było stworzenie sytuacji całkowitej wymienialności danych pomiędzy różnymi programami dla Amigi lub nawet wymienialności danych pomiędzy oprogramowaniem pracującym na zupełnie różnych komputerach. IFF to po prostu pewien sposób zapisu różnego rodzaju danych pozwalający na bezproblemową wymianę pomiędzy różnymi programami.

Standard ten zyskał zresztą powszechną akceptację użytkowników - idea wymienialności danych zostala w pełni dochowana - to po prostu działa! Np. plik sampla zapisany w standardzie IFF uzyskany przy pomocy programu Audiomaster IV może zostać z powodzeniem wykorzystany w ProTrackerze czy OctaMEDzie.

Powszechnie wiadomo, że standard IFF to nie tylko dźwięk. Można przecież przy jego pomocy zapisać również inne dane (np. grafikę). Jak to się więc dzieje, że komputer potrafi odróżnić wczytywany plik danych i np. pod Deluxe Painta nie da się wczytać sampla zapisanego w końcu w tym samym standardzie. Otóż związane to jest ze szczególną budową plików standardu IFF. Określone grupy danych posiadają swoje standardowe etykiety, dzięki którym programy mogą je zidentyfikować, a następnie przyjąć lub odrzucić. Etykieta ta zawiera 4 znaki i dla 8-bitowego sampla są to: 8SVX (8-bit Sampled Voice data). Możliwe jest tworzenie różnych rodzajów takich plików włączając pliki monofoniczne, stereofoniczne, wielooktawowe, a nawet pliki zawierające wiele punktów zapętlenia. Cały plik sampla zapisanego w omawianym standardzie w dużym uproszczeniu zawiera następujące elementy: etykietę, rozmiar i dane. Zagłębiając się w strukturę plików formatu IFF, okazuje się, że są one nieco bardziej złożone. Na początku każdego pliku zawsze występuje blok FORM, który służy do identyfikacji pliku, stąd pierwsze 4 bajty zawierają właśnie tą nazwę. Następnie znajduje się etykieta 8SVX oznaczająca dane przesamplowanego dźwięku, informacja o jego rozmiarze i dwa bloki nazwane: VHDR (Voice HeaDeR) i BODY zawierające odpowiednio informacje dodatkowe oraz w końcu dane samego sampla. (Patrz Rysunek 2).

<- Budowa pliku zawierającego dane dźwięku zapisanego w standardzie IFF ->
<- Etykieta FORM 8SVX ->
<- ROZMIAR ->
<- Nagłówek VHDR ->
<- Dane BODY ->



Rysunek 2.

Informacje zawarte w bloku VHDR dotyczą przeważnie sposobu odegrania sampla. Można więc tworzyć sample zwykłe - odgrywane tylko jeden raz oraz sample ciągłe - odgrywane 'w kółko' od początku do końca. Są one bardzo często wykorzystywane przez muzyków zajmujących się komponowaniem muzyki do gier i programów demonstracyjnych. Dzięki zastosowaniu tego typu instrumentów uzyskać można efekt ciągłego brzmienia np. instrumentu klawiszowego stanowiącego tło dla pierwszoplanowego sampla (najczęściej o krótkim czasie wybrzmiewania). Dzięki takiemu rozwiązaniu zys



Rysunek 3.

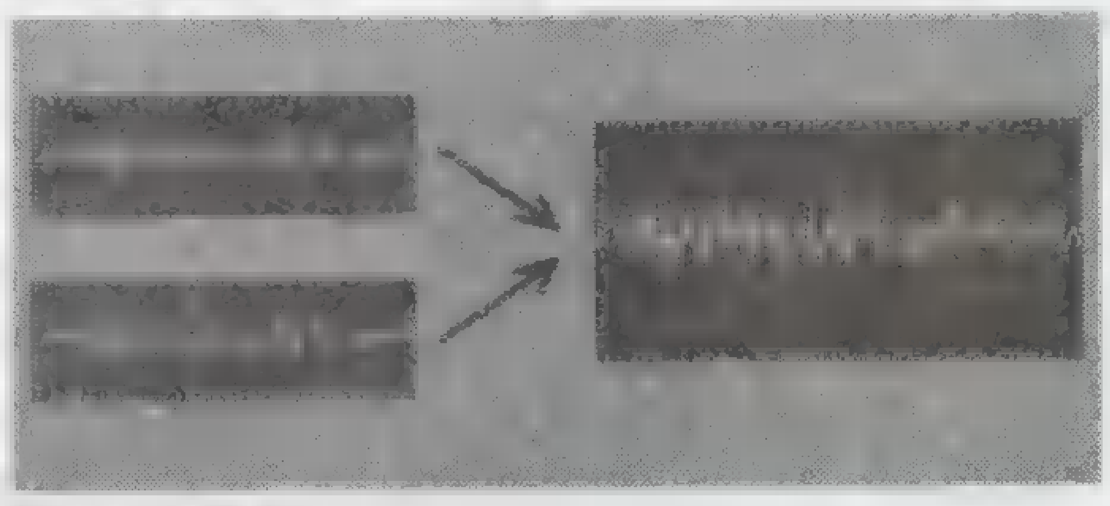
kujemy wiele na pamięci, gdyż przy odrobinie wprawy, nawet przy zastosowaniu bardzo krótkich próbek, możemy uzyskać dźwięk stale wybrzmiewający. Większość oprogramowania służącego do edycji sampli (np. Audiomaster IV) udostępnia jeszcze jeden sposób wielokrotnego odgrywania sampla, a mianowicie użytkownik posiada możliwość zaznaczenia wielu kolejno odgrywanych fragmentów sampla. Pozwala to na tworzenie bardzo długich fragmentów muzyki przy znacznej oszczędności pamięci. Jeżeli chcesz np. przesamplować jakiś utwór muzyczny składający się ze stale powtarzających się i przeplatanych ze sobą motywów, możesz pokusić się o przesamplowanie tylko fragmentów znacznie się od siebie różniących i odgrywanie ich w odpowiedniej kolejności. Jest to co prawda nieco pracochłonne, ale efekt wart jest wysiłku. (Patrz Rysunek 3).

Przy pomocy standardu IFF zapisywać można również sample stereofoniczne. Zasadniczo ich struktura jest identyczna jak sampli monofonicznych z tą jednak różnicą, że posiadają one jakby dwie części zawierające przesamplowane dane (oddzielnie dla kanału lewego i prawego).

<- Budowa bloku BODY sampla stereo ->
<- dane sampla dla lewego kanału audio ->
<- dane sampla dla prawego kanału audio ->

Sample stereofoniczne brzmią całkiem nieźle, posiadają jednak jedną zasadniczą wadę - pochłaniają dwukrotnie więcej miejsca w pamięci (i na dyskietce) co sample mono. Poza tym podczas konwersji sampla stereofonicznego z formatu IFF na RAW występuje jeszcze jedno zadziwiające, na pierwszy rzut oka (a właściwie ucha), zjawisko: sampl zostaje odegrany dwukrotnie. (Patrz Rysunek 5).

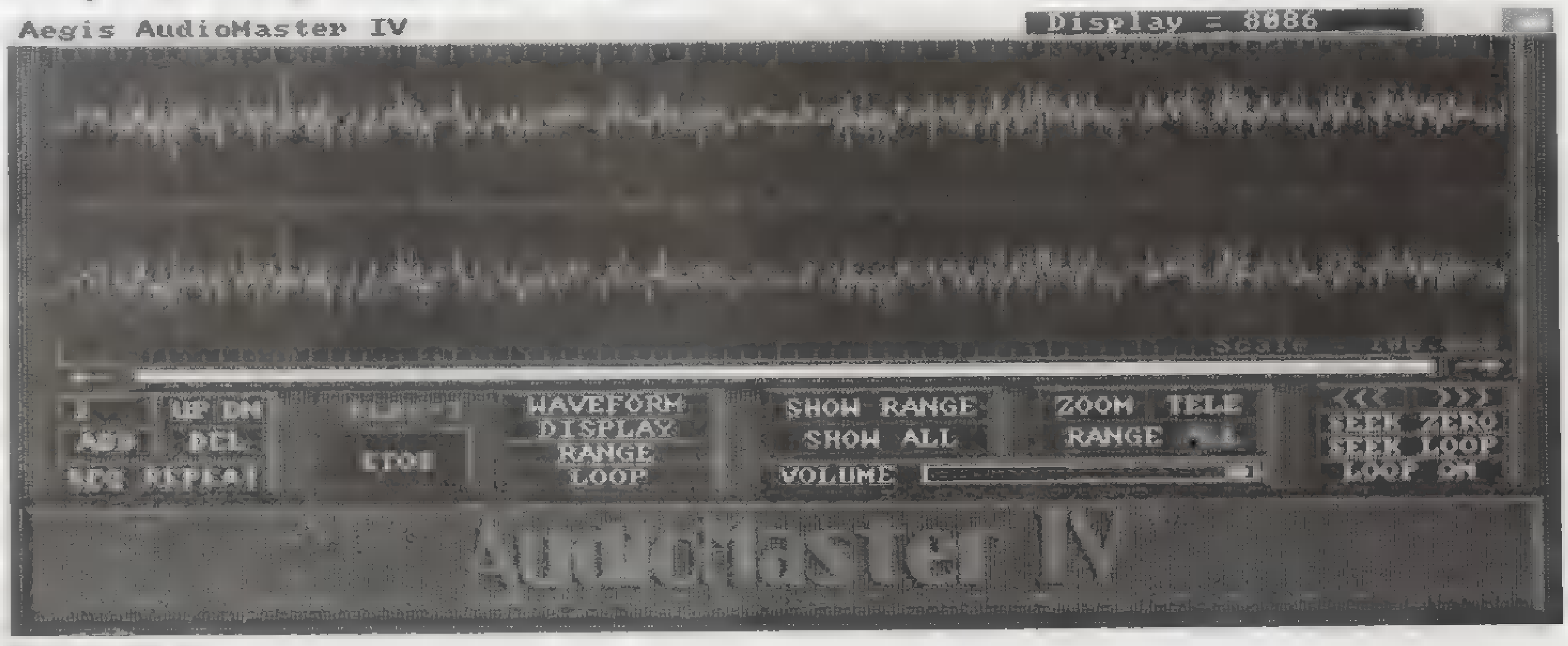
Dlaczego tak sie dzieje? Związane jest to ze specyfiką standardu RAW, który jest znacznie prostszy od wcześniej omawianego. Zawiera on jedynie dane przesamplowanego dźwieku bez żadnych informacji dodatkowych. Nie posiada więc żadnego nagłówka, dzięki któremu program mógłby w jakikolwiek sposób rozpoznać jego strukturę. W czasie konwersji z IFF na RAW na-


Rysunek 5.

stępuje odrzucenie nagłówka, a przez to utracenie wszystkich informacji dodatkowych na temat sampla. Program nie rozpoznaje więc, że dźwięk jest stereofoniczny i zwyczajnie odgrywa kolejno po sobie występujące sample (przeznaczone pierwotnie odpowiednio dla lewego i dla prawego kanału audio). Jako że różnice dźwięku dla lewego i prawego kanału są zwykle niewielkie, nie jesteśmy w stanie ich wychwycić, szczególnie, że nie mamy możliwości odsłuchania obu fragmentów jednocześnie. Ostatecznie więc, słyszymy dwa identyczne występujące po sobie sample. Efekt ten występuje oczywiście podczas wykonania najprostrzej konwersji i można się jego wystrzec wykorzystując profesjonalne oprogramowanie (AudioMaster itp.). Podczas dokonywania konwersji (nawet przy pomocy dobrego oprogramowania) tracimy również wszystkie informacje dotyczące zapętlenia dźwięku bez możliwości ich odzyskania. Warto więc o tym pamiętać i podczas dokonywania konwersji standardów zachować szczególną ostrożność.

Jako, że standard RAW nie umożliwia niczego więcej jak tylko zapis 'czystych' danych o dźwięku - nie widzę sensu dalszego rozpisywania się na jego temat. Powróćmy zatem do naszego pupilka - standardu IFF, który dzięki swojej strukturze posiada wielkie możliwości dalszego rozwoju. Zresztą rozwój ten następuje przez cały czas. Przykład? Proszę bardzo: Jeszcze do niedawna oprogramowanie nie umożliwiało zaznaczania wielu kolejno odgrywanych fragmentów sampla. Opcję tę wprowadzono po raz pierwszy dopiero w Audio Engineerze II i AudioMasterze IV. Tak więc próby wczytania takiego nowego rodzaju IFF-a do starszej wersji tych programów okazać się może do pewnego stopnia fiaskiem. Nowe informacje zawierające dane o kolejności i miejscu kolejnych punktów zapętlenia zostaną, po prostu, pominięte i sampl zostanie wczytany jako zwykły - jednokrotnie odgrywalny.

Ostatnio, wraz z nowymi lepszymi samplerami, pracuje się nad nowym usprawnieniem standardu IFF - możliwością zapisu sampli 16-bitowych. Kto wie, może za kilka lat pojawią się samplery 32-bitowe i konieczne będzie kolejne usprawnienie standardu! ❑



Rysunek 4.



Przyjaciółko, pokaż co masz.

Badanie konfiguracji systemu to jedna z najważniejszych czynności wykonywanych przez program. Od tego, w jaki sposób programista bada otoczenie pracującego programu zależy zachowanie kompatybilności przy przenoszeniu oprogramowania z jednego komputera na drugi.

Samo napisanie procedury testującej to jednak nie wszystko, ważne jest przede wszystkim to, w jaki sposób została ona napisana. Trochę głupią dla piszącego jest sytuacja, w której jego program nie "chodzi" na nowym komputerze z powodu... błędnego napisania procedury testującej konfigurację sprzętu. Prawdę mówiąc "trochę głupio" powinni się w tym momencie poczuć domorośli autorzy różnorakich dem, których zasięg działania ogranicza się częstokroć do konkretnego zestawu. Najczęstszym powodem występowania braku kompatybilności jest bezpośrednie odwoływanie się do różnych "tajemniczych" rejestrów pamięci zamiast do systemu operacyjnego. Ewentualna drobna zmiana w sprzęcie (typu nowszy scalak) zwykle prowadzi do błędnego działania. Z punktu widzenia programisty korzystającego z systemu nie następuje żadna zmiana, gdyż równocześnie ze zmianami sprzętowymi ulegają modyfikacji odpowiednie procedury systemowe. Odwołanie do takiej procedury zawsze da wynik taki, jak w poprzedniej wersji sprzętowej. Jasne, że systemy operacyjne się zmieniają, jednak zanim poszczególne wersje systemu przestaną być między sobą kompatybilne, minie znacznie więcej czasu.

Do podstawowych zadań programu zwykle należą operacje na plikach. Stąd przydatną rzeczą jest identyfikacja podłączonych stacji dysków, dysków twardych, itp. Jednakże zamiast odwoływań do urządzeń fizycznych (df0:, dh0:, ram:, itp.) lepiej odwoływać się do urządzeń logicznych (np. sys:, work:, itp). Odwołanie do urządzenia logicznego pozwala nam później na proste przenoszenie programu, bowiem wystarczy zmienić poleceniem ASSIGN ścieżkę przyporządkowaną danemu urządzeniu tak, by pasowała do nowego środowiska. Odnalezienie listy urządzeń fizycznych i logicznych jest tematem, na którym się skoncetrujemy.

Analizując strukturę DosLibrary możemy stwierdzić, że składowa dl_Root wskazuje na strukturę RootNode. Z kolei składowa tej struktury o nazwie rn_Info wskazuje na strukturę DosInfo. Jednakże wskaźnik rn_Info to wskaźnik BPTR! Wskaźniki te pochodzą z zamierzchłych czasów systemu operacyjnego Amigi, którego część została napisana w języku BCPL. Język ten jest poprzednikiem języka C i charakteryzuje się m.in. tym, że najmniejszy obiekt, który może zaadresować, to 32-bitowe długie słowo. Stąd wszystkie struktury, ciągi i inne dane, muszą znajdować się pod adresami będącymi krotnością 4 bajtów. Chcąc przekształcić wskaźnik BPTR w APTR lub inny "łapany" przez C należy pomnożyć jego wartość przez 4.

Ze struktury DosInfo mamy już niedaleko. Jej składowa di_DevInfo to wskaźnik (też BPTR!) na strukturę DeviceNode. Struktura ta ma następującą postać:

```
struct DeviceNode
{
  BPTR dn_Next;
  ULONG dn_Type;
  struct MsgPort *dn_Task;
  BPTR dn_Lock;
  BSTR dn_Handler;
  ULONG dn_StackSize;
  LONG dn_Priority;
  BPTR dn_Startup;
  BPTR dn_SegList;
  BPTR dn_GlobalVec;
  BSTR dn_Name;
}
```

Jak łatwo się domyślić, dn_Next wskazuje na następną strukturę DeviceNode (lub równy jest NULL w przypadku ostatniej struktury w liście). Składowa dn_Name wskazuje na nazwę urządzenia, natomiast składowa dn_Type określa jego typ: DLT_DEVICE, DLT_DIRECTORY, DLT_VOLUME. W przypadku wartości DLT_DEVICE, wartości struktury DeviceList opisują urządzenie fizyczne (df0:, dh0:, ram:), w przypadku DLT_DIRECTORY - urządzenie logiczne (tj. zdefiniowane poleceniem ASSIGN). Dla dn_Type równego DLT_VOLUME (urządzenie fizyczne identyfikowane według nazwy, np. "Workbench:", a nie "dh0:") traktuje się, że struktura DeviceNode przyjmuje postać struktury DeviceList:

```
struct DeviceList
{
  BPTR dl_Next;
  LONG dl_Type;
  struct MsgPort *dl_Task;
  BPTR dl_Lock;
```

```
        struct DateStamp dl_VolumeDate;
        BPTR dl_LockList;
        LONG dl_DiskType;
        LONG dl_unused;
        BSTR *dl_Name;
    }
```

Rzeczą godną odnotowania jest tu podobne rozmieszczenie niektórych składowych, np. dl_Next i dn_Next zajmują pierwsze 4 bajty, dl_Type i dn_Type kolejne 4 bajty, również dl_Name i dn_Name okupują ten sam obszar pamięci (bajty z offsetem 40-43). Tak więc nie ma specjalnego znaczenia, czy odwołując się do tych składowych informujemy kompilator, że chcemy dostępu do struktury DeviceNode czy DeviceList.

Dołączony listing programu pozwala zorientować się w sposobie odnajdywania urządzeń w liście systemowej. Program był kompilowany kompilatorami AztecC v5.2 i SAS-C v6.5. Efektem jego działania jest wypisanie wszystkich dostępnych aktualnie urządzeń, stąd powinien być uruchamiany z poziomu shella.

Na koniec jeszcze jedna uwaga. Oczekiwanie na włożenie określonej dyskietki do stacji realizuje się zwykle w pętli, która przeszukuje listę od początku do końca. Jednakże może wystąpić sytuacja, kiedy w trakcie przeskakiwania pomiędzy poszczególnymi elementami listy ulegnie ona modyfikacji (np. wkładamy lub usuwamy dysk). Taka sytuacja może być groźna, stąd bezpieczniej byłoby wyłączyć przerwania przed rozpoczęciem przeszukiwania, włączyć po zakończeniu oraz odczekać np. 1/5 sekundy (Delay(5*50)) i ponownie rozpocząć przeszukiwanie. ❑

```
#include <clib/exec_protos.h>
#include <libraries/dos.h>
#include <libraries/dosextens.h>
#include <libraries/filehandler.h>
#include <stdlib.h>
#include <stdio.h>


/* Wskaźnik na bibliotekę dos.library. */

extern struct DosLibrary *DOSBase;

main()
{
    ULONG brud;
    struct  RootNode *rootnode;
    struct  DosInfo *dosinfo;
    struct  DeviceList *devicelist;
    struct  DeviceList *pointer;
    struct  DeviceNode *devicenode;

    /* Otwarcie biblioteki dos.library. */

    if ( !( DOSBase = (struct DosLibrary *) OpenLibrary( (STRPTR)
        "dos.library", 0 ) ) )
    {
      printf( "Brak 'dos.library'!" );
      exit(0);
    }

    /* Wskaźnik na strukturę RootNode. */

    rootnode = (struct RootNode *) DOSBase->dl_Root;

    /* Wskaźnik na strukturę DosInfo. */
    brud = (ULONG) rootnode->rn_Info;
    brud *= 4;                          /* Konwersja wskaźnika. */
    dosinfo = (struct DosInfo *) brud;

    /* Wskaźnik na strukturę DeviceList. */

    brud = (ULONG) dosinfo->di_DevInfo;
    brud *= 4;                          /* Konwersja wskaźnika. */
    devicelist = (struct DeviceList *) brud;

    /* Rozpoczęcie przeglądania listy. */

    pointer = devicelist;
    printf( "Początek listingu:");

    while( pointer )
     {
        brud = pointer->dl_Name;
        /* Rozmieszczenie dl_Name i dn_Name pokrywa się. */
        brud *= 4;                      /* Konwersja wskaźnika. */
        brud++;
        printf( "%s:", brud );
        printf("  Typ = ");

        if ( pointer->dl_Type == DLT_DIRECTORY )
          printf("DIRECTORY");
        else
          if ( pointer->dl_Type == DLT_DEVICE )
            printf("DEVICE");
          else
            if ( pointer->dl_Type == DLT_VOLUME )
              /* Rozmieszczenie dl_Type i dn_Type pokrywa się. */
              printf("VOLUME");

        printf("  Task = %d", pointer->dl_Task );
        printf("  Lock = %d", pointer->dl_Lock );

        if ( pointer->dl_Type == DLT_VOLUME )
          {
            printf("  dl_LockList = %d", pointer->dl_LockList );
            printf("  dl_DiskType = %d", pointer->dl_DiskType );
          }
        else
          {
            devicenode = (struct DeviceNode *) pointer;
            printf("  dn_Handler = %d", devicenode->dn_Handler );
            printf("  dn_StackSize = %d", devicenode->dn_StackSize );
            printf("  dn_Priority = %d", devicenode->dn_Priority );
            printf("  dn_Startup = %d", devicenode->dn_Startup );
            printf("  dn_SegList = %d", devicenode->dn_SegList );
            printf("  dn_GlobalVec = %d", devicenode->dn_GlobalVec );
          }

        brud = pointer->dl_Next;
        brud *= 4;
        pointer = (struct DeviceList *) brud;
     }

  printf( "Koniec listingu!");

  /* Zamknięcie bibliotek. */

  if ( DOSBase )
    CloseLibrary( (struct Library *) DOSBase );
}
```

Odcinek dla BANKU

AMIGOWIEC

Zamówienie na prenumeratę i numery archiwalne

Zamawiam prenumeratę **AMIGOWCA** od numeru:	
Zamawiam prenumeratę **Dysków P.D.** od numeru:	

Zamówienie na numery archiwalne:

5/92	6/92	3/4	5/6	7/8	9/10

Zamówienie na dyskietki Public Domain:

2	3	4	5	6	7
8	9	10	11	12	A
B	C	1/93	2/93	3-4/93	5-6/93
7-8/93	9-10/93				

Odcinek dla posiadacza rachunku

AMIGOWIEC

Zamówienie na prenumeratę i numery archiwalne

Zamawiam prenumeratę **AMIGOWCA** od numeru:	
Zamawiam prenumeratę **Dysków P.D.** od numeru:	

Zamówienie na numery archiwalne:

5/92	6/92	3-4	5-6	7-8	9-10

Zamówienie na dyskietki Public Domain:

2	3	4	5	6	7
8	9	10	11	12	A
B	C	1/93	2/93	3-4/93	5-6/93
7-8/93	9-10/93				

Pokwitowanie dla wpłacającego

AMIGOWIEC

Zamówienie na prenumeratę i numery archiwalne

1. Aby zaprenumerować AMIGOWCA i / lub dyskietki Public Domain należy wpłacić na nasze konto:
   - 138 tys. za 6 wydań AMIGOWCA od dowolnie wybranego numeru.
   - 228 tys. za 6 dyskietek Public Domain od dowolnie wybranego numeru.

   Aby otrzymać numery archiwalne AMIGOWCA i / lub zamówić dyskietki PD wystarczy wpłacić na nasze konto:
   - 7 tys. za numery (5 i 6),
   - 16 tys. za numery (12),
   - 18 tys. za numery (3/4, 5/6, 7/8),
   - 23 tys. za numer (9-10)
   - 48 tys. za każdą dyskietkę PD.
2. Podać czytelnie swój adres.
3. Zaznaczyć na obu odcinkach blankietu, które numery zamawiamy.

W razie niejasności, czy jakichkolwiek opóźnień prosimy o informacje na adres:
P.W.H. "ALFIN" sp. z o.o.
tel. 28-79-20
ul. Świętojańska 2/7, 85-017 Bydgoszcz

Drodzy Czytelnicy!

Proponujemy półroczną prenumeratę naszych czasopism: AMIGOWIEC i ŚWIAT GIER KOMPUTEROWYCH rozpoczynającą się od dowolnie wybranego numeru. Można również zaprenumerować dyskietki Public Domain oraz zamówić archiwalne egzemplarze AMIGOWCA.

Wystarczy czytelnie wypełnić obie strony wybranego blankietu prenumeraty i wpłacić odpowiednią kwotę na podane konto.
Świat Gier Komputerowych w prenumeracie kosztuje taniej - tylko 23 tys. zł.
Również AMIGOWIEC jest tańszy w prenumeracie - tylko 23 tys. zł.
Dyskietki Public Domain są tańsze dla prenumeratorów naszych czasopism - tylko 38 tys. zł.

Odcinek dla BANKU

Świat Gier Komputerowych

Zamawiam prenumeratę na 6 numerów / na 12 numerów

od numeru:	od numeru:

Numery archiwalne

2	3-4	5-6	7-8	9-10	11-12

Odcinek dla posiadacza rachunku

Świat Gier Komputerowych

Zamawiam prenumeratę na 6 numerów / na 12 numerów

od numeru:	od numeru:

Numery archiwalne

2	3-4	5-6	7-8	9-10	11-12

Pokwitowanie dla wpłacającego

Świat Gier Komputerowych

Aby zaprenumerować Świat Gier Komputerowych, należy:

1. Wybrać rodzaj prenumeraty i zaznaczyć, od którego numeru ma się rozpoczynać.
2. Podać czytelnie swój adres.
3. Wpłacić odpowiednią kwotę na nasze konto, tj. 138.000,- za 6 wydań lub 276.000,- za 12 wydań.

Jednocześnie informujemy, że P.W.H. "ALFIN" ani ŚGK nie będą odpowiadać za problemy wynikłe z błędnego wypełnienia blankietu.

W razie niejasności, czy jakichkolwiek opóźnień prosimy o zgłaszanie się na adres:

P.W.H. "ALFIN" sp. z o.o.
tel. 28-79-20, fax 22-64-03
ul. Świętojańska 2/7,
85-017 Bydgoszcz

Pokwitowanie dla wpłacającego

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla posiadacza rachunku

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla BANKU

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

AMI-MARKET

Wystarczy wypełnić DRUKOWANYMI literami zamieszczony obok blankiet. Ogłoszenia mogą zamieszczać tylko osoby prywatne. Ważny jest tylko oryginalny, wycięty z czasopisma blankiet (NIE jego kserokopia!). Ogłoszenia dotyczące sprzedaży muszą zawierać cenę. Uwaga! Prosimy zaznaczyć czasopismo, w którym ma się ukazać ogłoszenie:

❒ AMIGOWIEC
❒ Świat Gier Komputerowych

ZAPRASZAMY

imię nazwisko

miejsce zamieszkania

kod miejscowy

telefon, fax

UWAGI

BŁYSKAWICZNE DARMOWE OGŁOSZENIA sprzedam, kupię, wymienię

Pokwitowanie dla wpłacającego

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla posiadacza rachunku

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

Odcinek dla BANKU

zł ....

słownie

WPŁACAJĄCY
imię
nazwisko

ADRES

kod pocztowy

na rachunek:
Bank PKO S.A. w Bydgoszczy
5.09011-400933.9-2511-30-111.0

P.W.H. "ALFIN" sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz

Opłata

zł ....

podpis przyjmującego

AMI-MARKET

Sprzedam literaturę komputerową za 30% wartości. Spis - koperta zwrotna. **Czesław Czyżyk, ul. 1-go Maja 1/3, 58-520 Janowice Wielkie.**

Sprzedam Amigę 600, gwarancja do maja 1994 r. Cena ok. 4.1 mln zł. **Maciej Cwir, ul. Juranda 3/7, 10-064 Olsztyn, tel. 23-70-01 (po południu).**

Kupię magazyn "Amiga" nr: 3/93, 5/93, 6/93, 7/93. Oferty z ceną na adres: **Cezary Pyka, ul. Błogosławionej Jadwigi Królowej 12/5, 97-500 Radomsko.**

Nowa gdańska grupa Inquisition poszukuje grafika i muzyka. Propozycje przesyłać na adres: **Piotr Kaźmierczak, ul. Władysława IV 7/15 b.19, 81-353 Gdynia, tel. (058) 216-107 lub (058) 204-219.**

Sprzedam A500 z wyposażeniem (joystick, modulator, 40 dyskietek). Cena 3.5 mln zł. Rozszerzenie 2 MB wewnętrzne do A500 (gwarancja)-1.3 mln zł. **Stawski Grzegorz, 87-400 Golub-Dobrzyń, tel. 34-15.**

Kupię A1200 nie drogo. Może być używana. **Kubera Krzysztof, ul. Dębowa 6/13, 98-100 Łask (woj. sieradzkie).**

Pilnie sprzedam A600 (gwarancja do 10/94), 90 dyskietek z pudełkiem. Wszystko za 6.5 mln zł. Stan idealny. **Kubera Krzysztof, ul. Dębowa 6/13, 98-100 Łask (woj. sieradzkie).**

Sprzedam komputer Amiga 500, pudełko na dyski, dyskietki, literatura. Cena ok. 5 mln zł. Komputer ma 1 MB pamięci. Pilne. **Bartłomiej Muszyński, ul. P. Skargi 10/32, 87-140 Chełmża (woj. toruńskie), tel. 75-31-04.**

Sprzedam Amigę 500 z pokrywą. Cena 4 mln zł. **Andrzej Czerwiński, ul. Ks. Londzina 40, 43-246 Strumień.**

Sprzedam oryginalną grę "The Immortal". Cena: 120 tys. zł. Wysyłka za zaliczeniem pocztowym. **Tomasz Kuna, oś. XXX Lecia 127/17, 44-286 Wodzisław Śląski, tel. 55-57-51.**

Kupię program do Packet Radia AHP v.3.0. Oferty z ceną proszę kierować na adres: **Robert Izdebski, ul. Sportowa 2/27, 63-200 Jarocin.**

Sprzedam Amigę 500 (1 MB Chip, 2 MB Fast-Megamix), Action Replay Mk3, dyskietki, literaturę. Cena 7.5 mln zł. **Błażej Wadelski, ul. Nowiny 2/4, 62-506 Konin 8.**

Sprzedam Amigę 500, 1 MB, joystick, modulator, komplet przewodów, ok. 190 dysków, 2 pojemniki na dyski, miesięczniki. Cena: 6 mln zł. **Andrzej Jukowski, ul. Modrzewiowa 15, 49-300 Brzeg, tel. 26-81 (po godz. 16).**

AMI-MARKET

TV-monitor Westa 14", pilot, RGB. Sprzedam za 3 mln zł. **Krzysztof Chwistek, ul. 11 Listopada 63/7, 43-300 Bielsko Biała.**

Kupię: "Amigowiec" 1, 10 czarno-białe, "Ami ga" 0, 3, 4/92, "Kebab"-wszystkie. **Wiesław Gawron, ul. Krakowska 1/37, 44-335 Jastrzębie Zdrój, tel. (032) 719-911.**

Kupię schemat monitora 1084S. **Jerzy Dziadur, ul. Konarskiego 21a/3, 30-049 Kraków.**

Korespondencyjny Klub Amiga. Oprogramowanie, porady. Info-koperta i znaczek. KKA, **skrytka pocztowa 32, 27-100 Iłża, tel. 128.**

Wymienię sample do "Protrackera", fonty i tła do "Scali" oraz inne. **Dariusz Andrzejak, oś. Przyjaźni 20/162, 61-687 Poznań.**

Video monitor 1802 do C-64 sprzedam za 2.85 mln zł. **Marcin Kotynia, ul. Przełajowa 17, 94-045 Łódź, tel. 86-96-08.**

Sprzedam C-64 (lub zamienię na PC/386 SX) monitor mono, magnetofon, 500 programów, literaturę, 2 moduły, joystici. **Przemek Mielcarek Goraj 11a, 66-342 Goraj, tel. 93.**

Sprzedam Atari 800XL+stacja dysków California Acces 2001+magnetofon XC12+110 dysków+joystick. Cena: 3.5 mln zł. **Wojtek Gawinek, ul. Batalionu Zośka 7/8, 01-130 Warszawa, tel. 37-98-92 (dzwonić po godz. 18).**

Kupię stację dysków 5.25" do Amigi 600, może być używana. **Paweł Krzywdziuk, ul. Puławska 4/6/8m6, 02-566 Warszawa, tel. 49-31-18.**

Sprzedam Video Backup System. Nowy-gwarancja. Cena: 550 tys. zł. **Mariusz Walas, ul. Św. Ducha 88/III/21, 88-100 Inowrocław, tel. 732-53.**

Sprzedam Amigę 500, 2 MB, 30 gier, joystick, myszka, monitor 1084S, literatura. **Patryk Kupis, ul. Wojska Polskiego 96/24, 91-809 Łódź, tel. 55-52-44.**

Sprzedam C-64, magnetofon, stację dysków, dwa moduły, nowy joystick, dużo gier. Cena 3.6 mln zł. **Adam Wąsikowski, ul. S.K.Wyszyńskiego 7 bl.16/474, 05-220 Zielonka, tel. 611-91-80 wew. 723.**

Sprzedam Amigę 500 plus 2 MB Chip RAM, modulator, pokrywa, literatura, oprogramowanie. Cena do negocjacji od 5.6 mln zł. **Tomasz Banasik, ul. Nosala 1/28, 32-514 Jaworzno, tel. (035) 63-642.**

Sprzedam C-64, magnetofon, joystick, cartridge, gry, literaturę (cena ok. 2 mln zł) lub zamienię na Amigę, dodając motorynkę. **Tomasz Żuchowski, ul. Wilsona 37, Łódź.**

Sprzedam bardzo tanio oryginalną instrukcję w języku angielskim do Amigi 500 i "Deluxe Painta II" (30 tys. zł) oraz komiksy i ciekawe książki przygodowe dla 10-15 latków. Dla zainteresowanych - pełen katalog. **Piotr Zieliński, ul. Sikorskiego 70/4, 18-100 Łapy.**

Kupię i wymienię dyski pod emulator AMAX-II. **K.Chwistek, ul. 11 Listopada 63/7, 43-300 Bielsko-Biała.**

Sprzedam: C-64I z osprzętem i oprogramowaniem za 1.7 mln zł. Stację dysków 1541II+60 dysków za 1.7 mln zł. Wszystko razem za 3.3 mln zł. Stan bardzo dobry. **Maciej Lipiński, ul. Kopernika 2/10, 63-400 Ostrów Wielkopolski, tel. 36-63-39.**

Sprzedam C-64C (półroczny, stan idealny) za 1.2 mln zł, joysticki, dyskietki, kasety, Final 3, pudełka na dyski. **Piotr Radecki, ul. Złotkowska 35, 62-570 Rychwał, tel. 117.**

Sprzedam stację 1541 (stan bardzo dobry) za 1.2 mln zł, Final 3 za 120 tys. zł, dyskietki i kasety. **Piotr Radecki, ul. Złotkowska 35, 62-570 Rychwał, tel. 117.**

Sprzedam komputer C-64, magnetofon, 2 cartridge, joystick, 400 gier, literaturę. Całość za 1.8 mln zł. **Piotr Nowakowski, ul. Farna 7, 86-010 Koronowo, tel. 822-787.**

Sprzedam C-64II, 1541II, magnetofon, Final III, monitor 1802D (nie używany), joystick, 30 dyskietek, pudełko na dyskietki. **Łukasz Dąbrowski, ul. Okęcka 7/47, 02-658 Warszawa, tel. 43-71-58.**

Sprzedam A500 1MB, modulator, mysz, joystick, pokrywa, programy, 150 dyskietek, literatura, czasopisma. Cena: 6 mln zł. **Robert Sokołowski, ul. Sienkiewicza 71/9, 90-057 Łódź.**

Sprzedam A500, roczną (po naprawie jest na gwarancji do 06.94), 1 MB z przełącznikiem: za 3.5 mln zł lub dodatkowo 200 dyskietek i 2 pudełka, joystick, mysz: całość za 5 mln zł. **Michał Łuczak, ul. Sienkiewicza 89, 05-082 Babice k. Warszawy, tel. 36-84-17 wew. 421.**

Sprzedam C-64 z magnetofonem, stację dysków, joystick, 60 dyskietek z grami, Final II. Komplet za 2.5 mln zł. **Andrzej Filipczuk, ul. Zamiejska 6/1, 80-036 Gdańsk.**

Wymienię komputer "Nintendo" oraz cartridge: "Turtles 2 i 3", "Pinball", "150 un 1" itp. na Amigę 500. **Marcin Poręba, ul. Łączna 25a/4, 64-920 Piła, tel. 12-25-04.**

Kupię czasopisma "Computer Studio". **Artur Baćmaga, ul. Sucha 34, 26-681 Orońsko (Krogulcza).**

Sprzedam C-64, magnetofon, 2 cartridge: Black Box v.5 i v.8, 700 gier na 15 kasetach, literatura. Cena: 1.9 mln zł (mogę opuścić). **Michał Tołysz, ul. Toruńska 52d/36, 86-050 Solec Kujawski, tel. 87-19-08.**

Sprzedam CPC 464, monitor (zielony), 30 kaset gier, 2 joysticki, literaturę po polsku i niemiecku. Cena: 3.5 mln zł. **Stryczek Michał, ul. Krakowska 210/1, 40-394 Katowice 14, tel. 156-83-58.**

Kupię monitor kolorowy, używany. Oferty z ceną na podany adres lub telefonicznie. **Piotr Kujawski, ul. Wareńska 4a/25, 80-288 Gdańsk, tel. 48-55-89.**

A1200, 2 zewnętrzne stacje dysków, monitor 1084S, 200 dysków, czasopisma, 2 myszy, filtr, joystick. Cena: 14 mln zł. **Dariusz Bzdyra, ul. Buczka 7/8, 56-120 Brzeg Dolny (woj. wrocławskie), tel. 195-782.**

Sprzedam A500, 1MB, modulator, pokrywa, literatura, oryginalne programy: "Global Effect", "Bank Danych". Cena: 5 mln zł. **Robert Studziński, ul. Królowej Jadwigi 39/53, 33-300 Nowy Sącz.**

Sprzedam C-64, magnetofon, monitor Philips, joystick, 300 gier, cartridge. Cena 2.5 mln zł. **Mariusz Chrubczyński, ul. Wincentego 89/105, 03-291 Warszawa, tel. 11-85-14.**

Akcesoria do komputerów AMIGA

Amiga 500/500+	
Mega RAM 2 MB pamięci	4.180.000.-
Mega RAM 0 MB pamięci	1.690.000.-
Mega RAM HD 2 MB pamięci	5.280.000.-
Mega RAM HD 0 MB pamięci	2.790.000.-
AT-BUS HD - kontroler dysku twardego	1.690.000.-
Amiga 500/600/1200/2000	
Sampler Mono (40 kHz)	350.000.-
Sampler Stereo (30 kHz)	475.000.-
Midi interface (1 IN, 1 OUT, 1 THRU)	350.000.-
Amiga CDTV	
Interface do dwóch zwykłych joysticków i myszki	350.000.-
Amiga 1200	
2,5" lub 3,5" Hard Disk Kit - twardy dysk w środku komputera za niższą cenę (kable łączące, instrukcja i oprogramowanie w języku polskim)	390.000.-
E 1204 0 MB RAM	2.990.000.-

podane cene zawierają 22% VAT

Sprzedaż za zaliczeniem pocztowym po doliczeniu kosztów przesyłki.

Dla odbiorców hurtowych rabat do 30%.

Możliwość zakupu na raty.

MEGA RAM HD

zewnętrzne rozszerzenie do komputerów Amiga 500 i 500+

* powiększa pamięć komputera o 2, 4 lub 8 MB
* umożliwia dołączenie dysku twardego typu AT - bus
* dołączane zewnętrznie - gwarancja komputera pozostaje zachowana
* obudowa z tworzywa sztucznego, kolorystycznie i stylistycznie dopasowana do komputera
* autokonfigurujące i autobootujące
* możliwość powiększania pamięci - moduły SIMM 1 lub 4 MB, 120 ns lub szybsze
* "przelotowe" (umożliwia dołączenie np. cartridge'a)
* mały pobór mocy - nie jest potrzebny zewnętrzny zasilacz
* w pełni buforowane szyny danych i adresowa

E 1204

rozszerzenie pamięci do Amigi 1200

* podstawka pod kooprocesor arytmetyczny (68881/68882 max. 33 MHz)
* miejsce na 4 MB 32 bit FAST RAM (4 x SIMM 1 MB 3 CHIP 70 ns)
* zegar czasu rzeczywistego podtrzymywany akumulatorem

Dystrbutorzy naszych produktów:

AMIGA s.c.	Warszawa ul. Batorego 10, tel.: 25-60-31 w 103
AMIGO,	Wrocław ul. Leszczyńskiego tel.: 310-61 w. 294
bajPCtek,	Kraków ul. Wiślana 8 tel.: 22-59-72
	Nowy Sącz ul. Wąsowiczów 10
	Tarnóww ul. wałowa 2 tel.: 22-40-70
ALL IN ONE,	Łódź ul. Piotrowska 111 tel.: 327-324
INFO,	legnica ul. Złotoryjska tel.: 56-06-55
MIKROKOMPUTERY "XYZ" s.c.	Lublin ul. Okopowa 6, tel.: 21-394
MIKROMAN s.c.	Katowice Pl. Rostka 3, tel.: 515-132
	Rybnik ul. Wiejska 19, tel.: 233-56
	Bielsko-Biała Pl. Wolności 3, tel.: 229-70
PROABIT	Raszyn ul. Mickiewicza 14, tel.: 56-08-91
VADIM	Zielona Góra ul. Kupiecka 1, tel.: 656-72
XYZ - Mikrokomputery s.c.,	Lublin ul. Okopowa 6 tel.: 21-394

ELSAT s.c.
00-714 Warszawa
ul. Czerniakowska 28B
tel.: (22) 40-58-76
(2) 642-96-05
fax: (22) 41-52-80



ELSAT

Nazwy oraz znaki firmowe są zastrzeżone przez ich właścicieli.

EUREKA

* Soft- & Hardware *

dealer:

ALFA DATA

bsc

bsc büroautomation AG München

Problem:

Not enough memory... (A1200)
Potrzebny koprocesor, pamięć 32bit

Rozwiązanie:

Memory Master 1200,
karta rozszerzenia do Amigi 1200

Memory Master 1200 jest rozszerzeniem do Amigi 1200 wkładanym pod spód komputera "pod klapkę". Instalacja jest prosta i nie powoduje utraty gwarancji na komputer. W nowej wersji złącze karty jest od wewnątrz ścięte, co sprawia, że instalacja nie jest uciążliwa. Standardowa wersja zawiera 1Mb pamięci Fast i zegarek. Jest na niej miejsce na 8Mb 32-bitowej pamięci (moduły 0,5Mb w obudowach ZIP) oraz na koprocesor 68882 w obudowie PLCC do 50Mhz, a także na kwarc (oscylator). Rozszerzać można co 4Mb. Moduły pamięci są u nas w stałej sprzedaży (70ns) i są tańsze od popularnych modułów SIMM. Posiadamy w sprzedaży koprocesor 33Mhz.

Nowa wersja - w 100% kompatybilna z wszystkimi modelami A1200 (płytą 1.1B i 1.1D

Oferujemy produkty firmy AlfaData:
- skanery kolorowe (256 barw)
- myszki Mega Mouse 260 dpi, 400 dpi
- myszki bezprzewodowe
- trackballe
- pióra świetlne

066362714 HOTLINE

1 ROK GWARANCJI

polskie PL instrukcje

Stałe pozycje:
emulatory PC: ATOnce
KCS Power PC BOARD
kontrolery
At-Bus do A500 i A2000
rozszerzenia pamięci
do A500, A500+, A600
(także PCMCIA 2/4Mb)
Action Replay,
i inne hardware

Zapraszamy do naszych dystrybutorów:
Poznań: AVC - ul. Krysiewicza 5/3
Warszawa. ComputerProjekt ul. Smulikowskiego 1/3
Katowice: MICROMAN Pl. Rostka 3
Szczecin: Admirał ul. Monte Casino 37
Gdańsk Morena: Microserwis ul. Marusarzówny 6

EUREKA

62-300 Września ul. Wojska Polskiego 13
tel. (066)-362-714 godz. 9.00 - 16.00

Prowadzimy detaliczną sprzedaż wysyłkową.

50 km Poznań | Warszawa 256 km

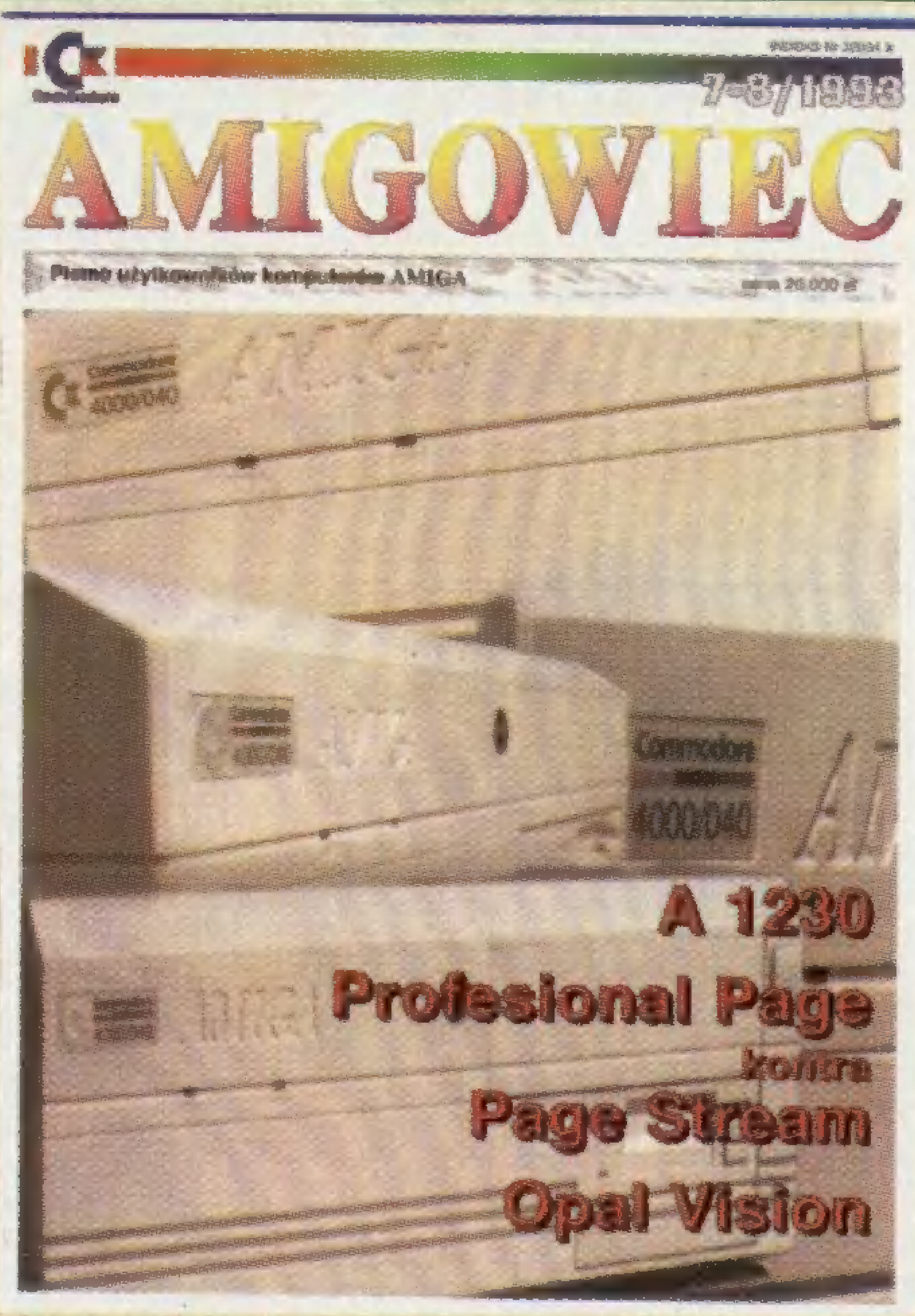
Wszystkie drogi prowadzą do Wrześni...

AMIGOWIEC 5-6/93



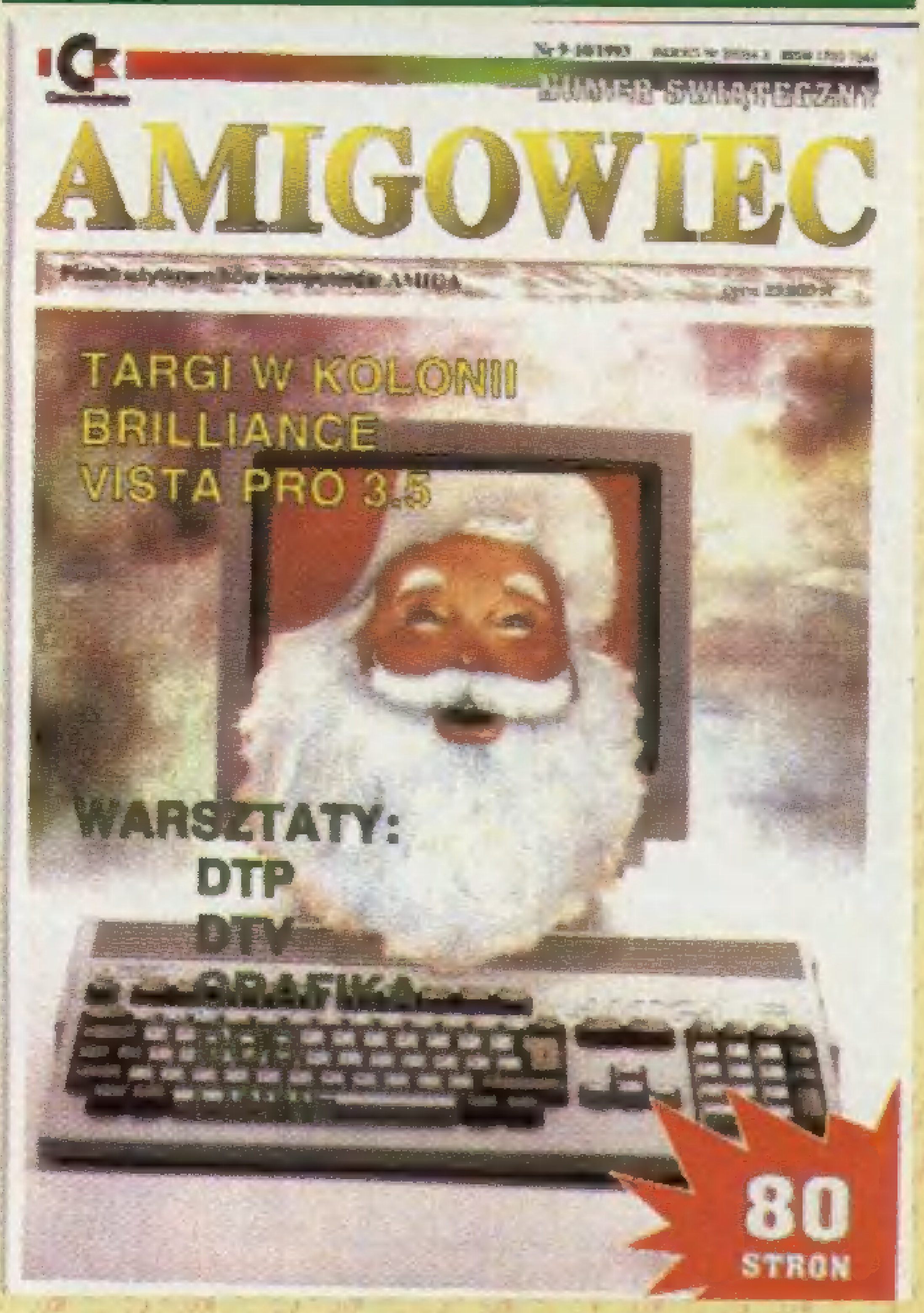
PD #5-6/93

AMIGOWIEC 7-8/93



PD #7-8/93

AMIGOWIEC 9-10/93



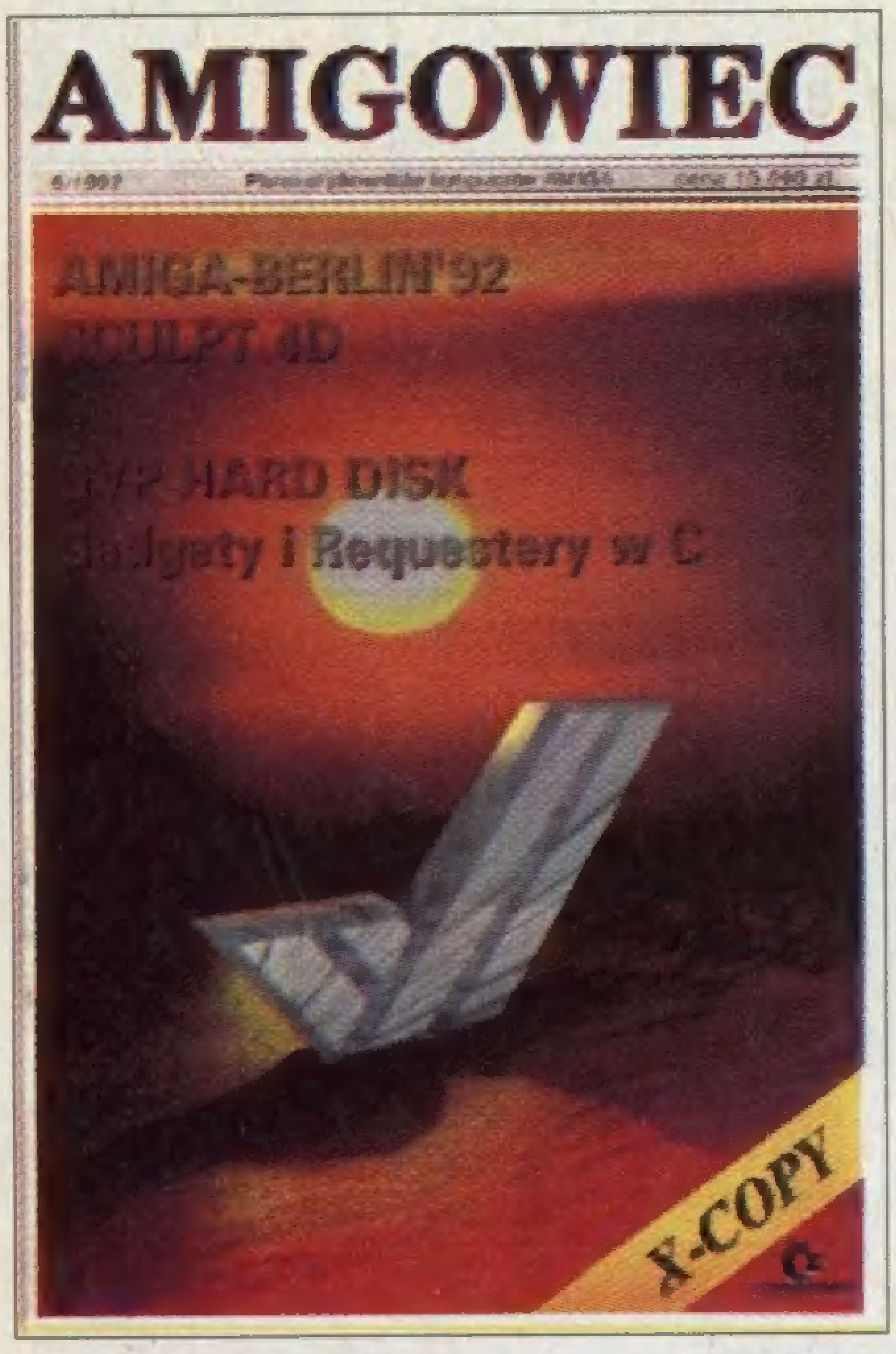
WARSZTATY

PD #9-10/93

Wszystkie przedstawione numery Amigowca z lat 92 i 93 są do nabycie po następujących cenach:

5/92 - 7 tys. zł.	3-4/93 - 18 tys. zł
6/92 - 7 tys.zł.	5-6/93 - 18 tys. zł
12/92 - 16 tys. zł.	7-8/93 - 18 tys. zł
1-2/93 - 18 tys. zł.	9-10/93 - 23 tys. zł.

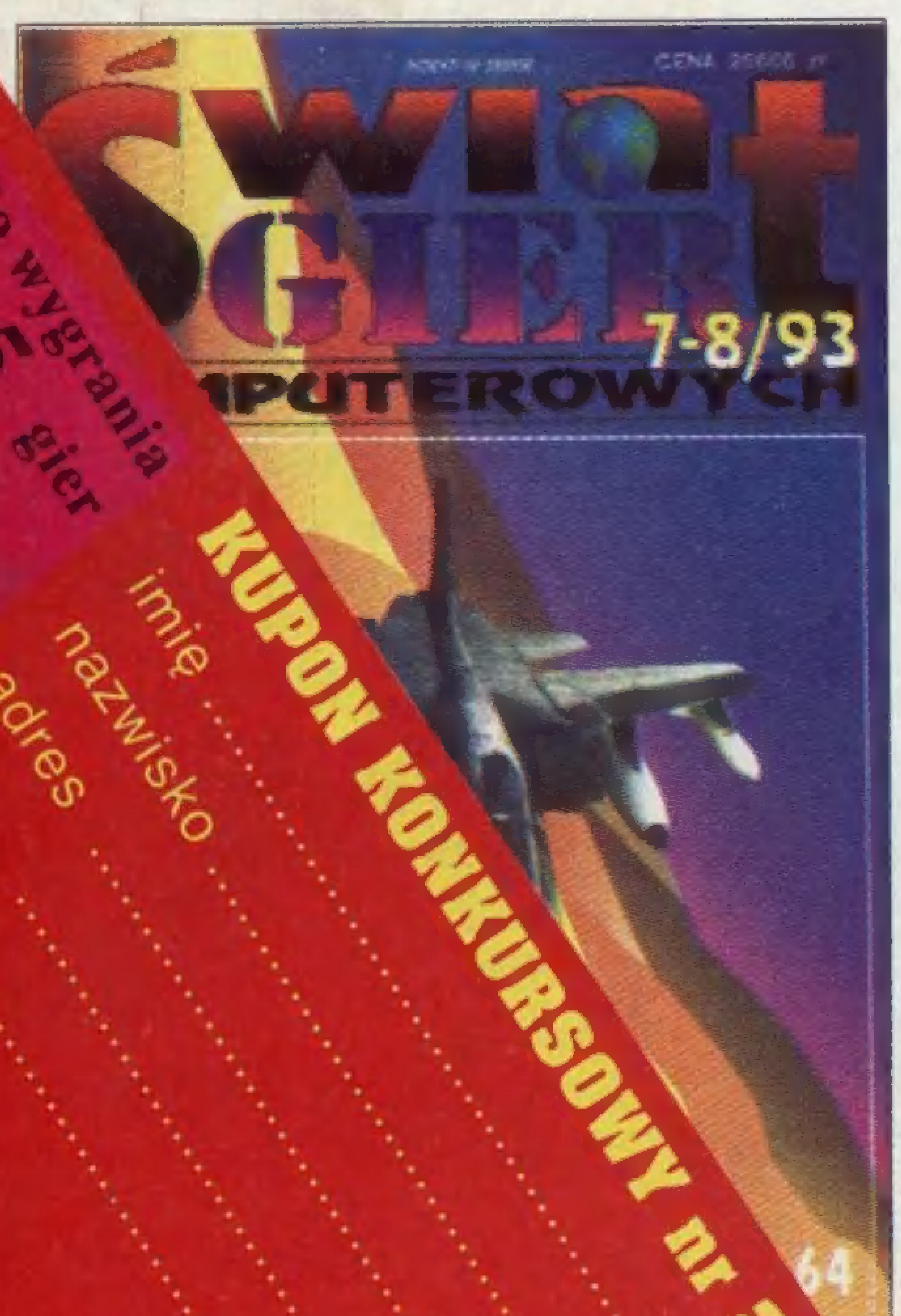
Informujemy także, że istnieje możliwość nabycia wszystkich dotychczas opublikowanych dysków PD Amigowca. Cena za jedną dyskietkę wynosi 40.000 + 22% VAT.

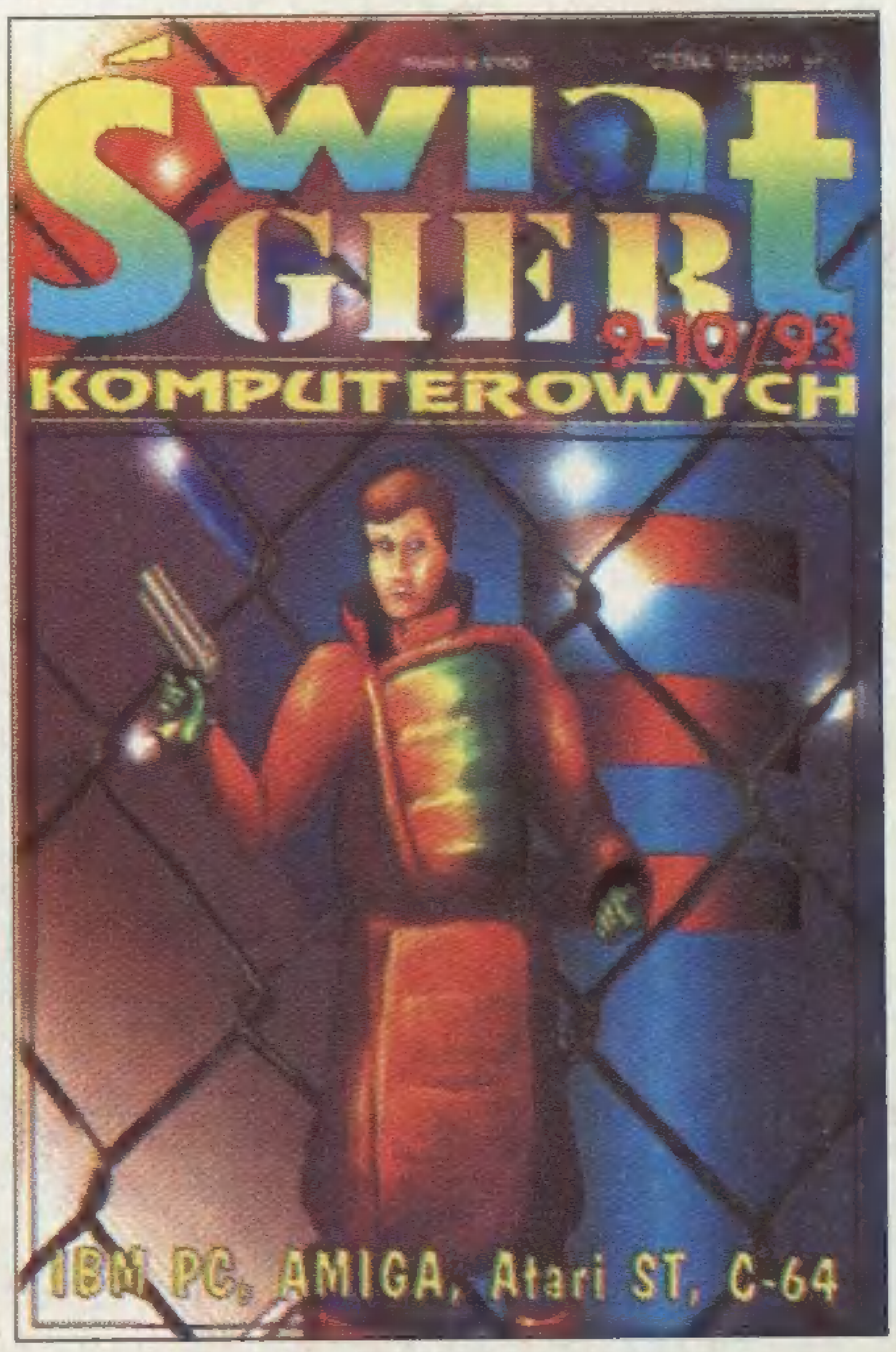












TEN ZESTAW MOŻE BYĆ TWÓJ ZA JEDYNE **80 tys. zł**

Czerwony kupon przesłany wraz z zamówieniem na zestaw na adres redakcji uprawnia do wzięcia udziału w losowaniu nagród (5 gier komputerowych)

+ AMIGOWIEC NIESPODZIANKA

Do wygrania 5 gier

KUPON KONKURSOWY nr 2

imię ....

nazwisko ....

adres ....

Swoje zamówienie wyślij na adres:

ALFIN
sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. 28-79-20
fax 22-64-03